

국가기술자격 기술사 시험문제

기술사 제 110 회

제 4 교시 (시험시간: 100분)

분야	정보통신	종목	컴퓨터시스템용용기술사	수험번호		성명	
----	------	----	-------------	------	--	----	--

※ 다음 문제 중 4문제를 선택하여 설명하시오. (각25점)

1. 대칭 암호화와 비대칭(공개키) 암호화를 각각 설명하고, 비교하시오.
2. 운영체제에서 문맥(context)의 내용, 문맥교환(context switch)의 과정 및 문맥교환이 발생하는 조건을 설명하시오.
3. DB 설계에 있어서 Primary Key와 Unique Index의 특성을 비교하고, Primary Key 사용 시의 장점에 대하여 설명하시오.
4. 통신망(Communication Network) 장치들의 물리적 구성 형태 및 각각의 특징에 대하여 설명하시오.
5. FDS(Fraud Detection System)의 개념, 구성요소 및 기능을 설명하시오.
6. 가상화 환경에서 데이터 백업방법 2가지를 설명하시오.

종 목	문 제
컴퓨터시스템응용기술사	4교시 1번

대칭 암호화와 비대칭(공개키) 암호화를 각각 설명하고, 비교하시오.

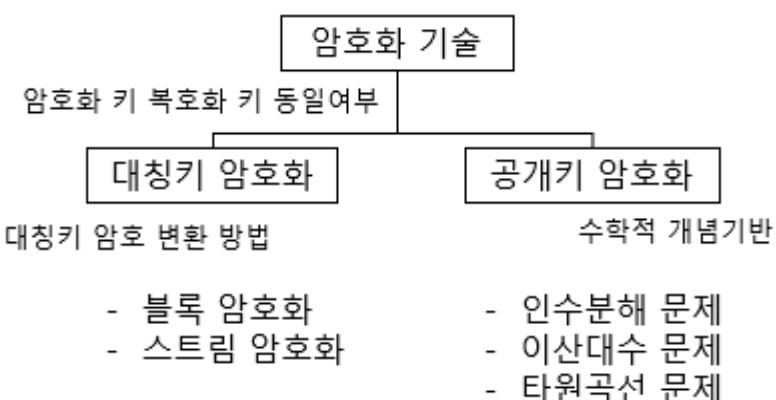
도메인/토픽	정보보안 > 암호화
키워드	암호화 복호화에 같은키 사용, 스트림 암호, 블록암호, 암호화(공개키)와 복호화(개인키)를 위해 2개 키사용, 디지털 서명,
풀이 가이드	<p>대칭 암호화와 비대칭(공개키) 암호화를 그림과 함께 큰 단락으로 설명 답안작성 시간 확보를 위해, 암호화에 대해 아는 지식이 많은 경우, - 1번은 가, 나 없이 암호화 분류만 하고, 2~4는 가, 나로 나누어서 대칭키암호화 와 공개키 암호하의 설명을 비교</p> <ol style="list-style-type: none"> 1. 암호화의 개요(분류, 필요성) <u>2. 대칭 암호화 설명</u> <u>3. 비대칭(공개키) 암호화 설명</u> <u>4. 대칭 암호화와 비대칭(공개키) 암호화 비교</u>
필수항목 /연관토픽	<p>암호학, 혼돈과 확산, 쌍방향 암호화, 대칭키, 공개키, 양자암호화, 단방향 암호화, 해시</p>
참고문헌	<p>정보보안 강의자료, 목포해양대학교 (http://lily.mmu.ac.kr/lecture/10is/4.pdf) 암호학, 나무위키 (https://namu.wiki/w/%EC%95%94%ED%98%B8%ED%95%99) RSA암호, 위키백과 (https://ko.wikipedia.org/wiki/RSA_%EC%95%94%ED%98%B8) 스트림암호, 위키백과 (https://ko.wikipedia.org/wiki/%EC%8A%A4%ED%8A%B8%EB%A6%BC_%EC%95%94%ED%98%B8)</p>
Advice	<p>암호학의 기본 이론 이해와 함께 쌍방향 암호화의 대칭키와 비대칭키(공개키)암호화 의 그림을 통한 설명이 가능해야 합니다. 막상 설명을 하려고 하면 설명이 막히는 경우가 많습니다. 토픽 이해여부에 대해 확실한 메타인지(MetaCognition)가 필요합니다. (3자에게 설명 가능 여부가 핵심) 양자암호화에 대한 학습하여 이후 나올 토픽에 대한 준비도 필요합니다.</p>
작성자	108회 컴퓨터시스템응용기술사 / 김윤경 (winnniw@naver.com)

1. 암호화(Encryption)의 개요

가. 암호화(Encryption)의 필요성

기능	역할	효과
- 정보시스템이 요구하는 정보의 보안 수준에 따라 효율적이고, 계층적인 보안 대책을 제공 - 정보시스템의 기밀성을 위해 암호화 기술을 이용	- 암호화 기술 이용: 전자화폐, 전자송금, 전자지갑 등에서 전자상거래의 신뢰성과 비밀성을 제공하는 역할	- 외부침입자(intruder)에 의한 보안 위협 예방 효과

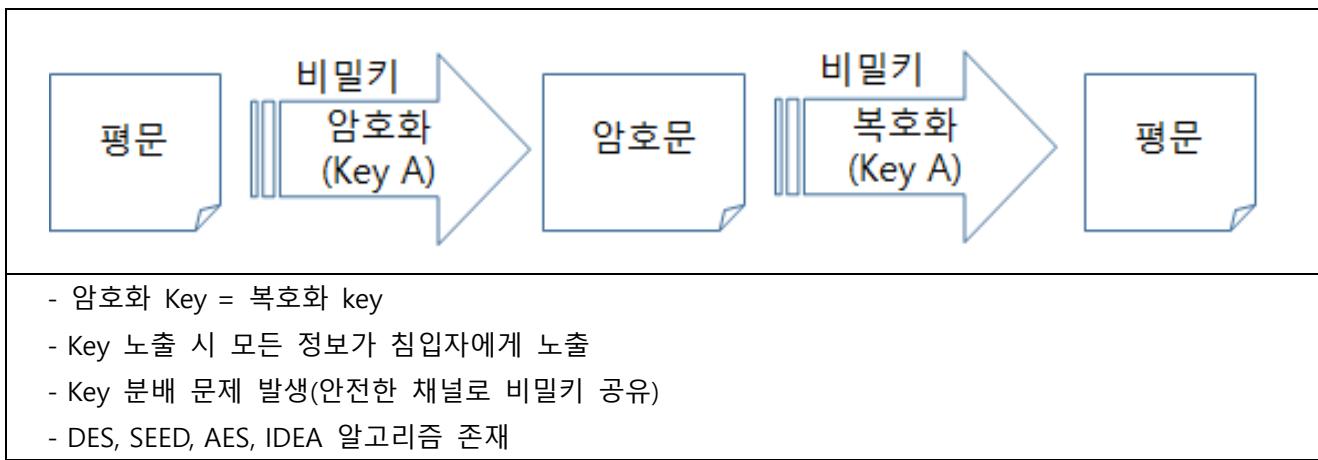
나. 암호화(Encryption)의 분류



- 암호화 키와 복호화 키가 동일한지에 따라 대칭키 암호화와 비대칭키(공개키)암호화로 분류

2. 대칭 암호화(Symmetric Cryptographic) 설명

가. 암·복호기가 동일한 단일키 방식 암호화, 대칭 암호화의 동작 원리



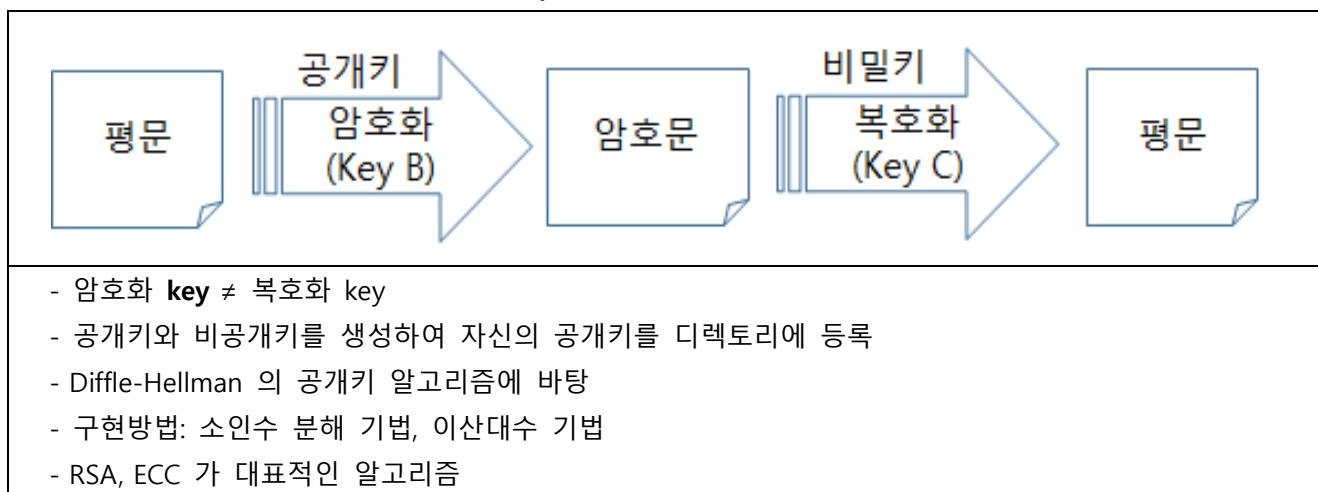
나. 대칭 암호화의 알고리즘 유형

유형	구조	설명	알고리즘
블록암호화	Feistel	- 암호화 과정과 복호화 과정이 동일한 방식 - 평문 전체를 블록단위로 배열하고 블록을 동일한 크기로 분할	- DES - SEED
	SPN	- 샤논(Shannon)의 혼돈(confusion)과 확산(diffusion) 이론을 기반 - 암호화 과정과 복호화 과정이 달라 암호화의 역 과정을 고려해야 함	- AES - ARIA
스트림암호화	동기식 /자기 동기식	- 쉬프트 레지스터를 이용한 이진수열 발생기를 사용하여 입력되는 정보를 비트단위로 암호화 - 일회용패드(One-time Pad)의 이론적 특성으로 부터 연유됨	- RC4 - A5

- 암호화 키와 복호화 키가 동일하면 대칭키, 서로 다르면 비대칭(공개키) 암호화

3. 비대칭 암호화 (공개키, Asymmetric Cryptographic) 설명

가. 각기 다른 암·복호키를 통한 암호화, 비대칭 암호화의 동작 원리



나. 비대칭 암호화의 RSA(Rivest Shamir Adleman) 알고리즘 설명

구분	방법	내용
1. 키생성	서로 다른 소수 p, q	- 소수 선택
	$N = p * q$	- 두 소수 곱
	$\Phi(N) = (p - 1) * (q - 1)$	- 오일러 파이 함수
	$1 < e < \Phi(N)$	- $\Phi(N)$ 와 서로소인 임의수 e 선택 공개키: N, e
	$e * d \pmod{\Phi(N)} = 1$	- e ($\pmod{\Phi(N)}$)에 대해 곱의 역원 d 계산 개인키: N, d
2. 암호화	$c = m^e \pmod{\Phi(N)}$	평문 m 에 대한 암호화

3. 복호화	$m = c^d \pmod{\Phi(N)}$	암호문 c 에 대한 복호화
--------	--------------------------	------------------

- $e * d \pmod{\Phi(N)} = 1$ 인 곱의 역원 관계와 서로소 관계를 활용하여 곱셈(나머지 연산)을 활용 암호화 복호화

4. 대칭 암호화와 비대칭(공개키) 암호화 비교

가. 암호화키 측면의 비교

비교항목	대칭 암호화	비대칭 암호화
키의 관계	- 암호화 키 = 복호화 키	- 암호화 키 \neq 복호화 키
키의 수	- 단일키	- 복수키(Pair of Keys)
키의 유형	- 비밀키	- 비밀키, 공개키 한 개씩
키의 관리	- 키관리 복잡	- 디지털 인증서와 인증기관 필요
키의 전송	- 필요	- 불필요

- 암호화 대상 및 활용에 따라 대칭 및 비대칭 암호화 방식을 적용

나. 암호화 알고리즘의 장단점 측면의 비교

구분	대칭 암호화	비대칭(공개키) 암호화
장점	<ul style="list-style-type: none"> - 암호화 속도 빠름 - 알고리즘 다양 	<ul style="list-style-type: none"> - 키의 안전한 분배 및 키 관리가 용이 - 강화된 보안성과 편리함 - 전자서명 기법을 이용 - 부인방지(Non-Reputation)
단점	<ul style="list-style-type: none"> - 가장공격(Impersonation)에 취약 	<ul style="list-style-type: none"> - 암호화 및 복호화를 위한 처리 속도가 대칭키 암호 방식보다 느린 편

- IT기술의 발전에 따른 암호화 알고리즘의 해석 시간이 단축됨에 따른 암호해독 문제를 양자암호화의 양자중첩, 양자 얹힘 특징을 이용하여 대칭암호화와 비대칭 암호화(RSA)의 단점 보완 가능

[참고] 암호체계(Encryption System)의 기본 가정

- 암호체계는 완전히 공격자에게 알려져 있고, 단지 키만이 유일한 비밀이다.
- 커크호프 원칙(Kerckhoffs Principle): 암호 알고리즘은 비밀이 아니다.
- 비밀 알고리즘들은(공개되었을 때) 알고리즘 자체는 강력하지 않을 경우가 많다.
- 비밀 알고리즘은 결국 공개된다. 따라서, 사전에 취약점을 발견하는 것이 유리하다.

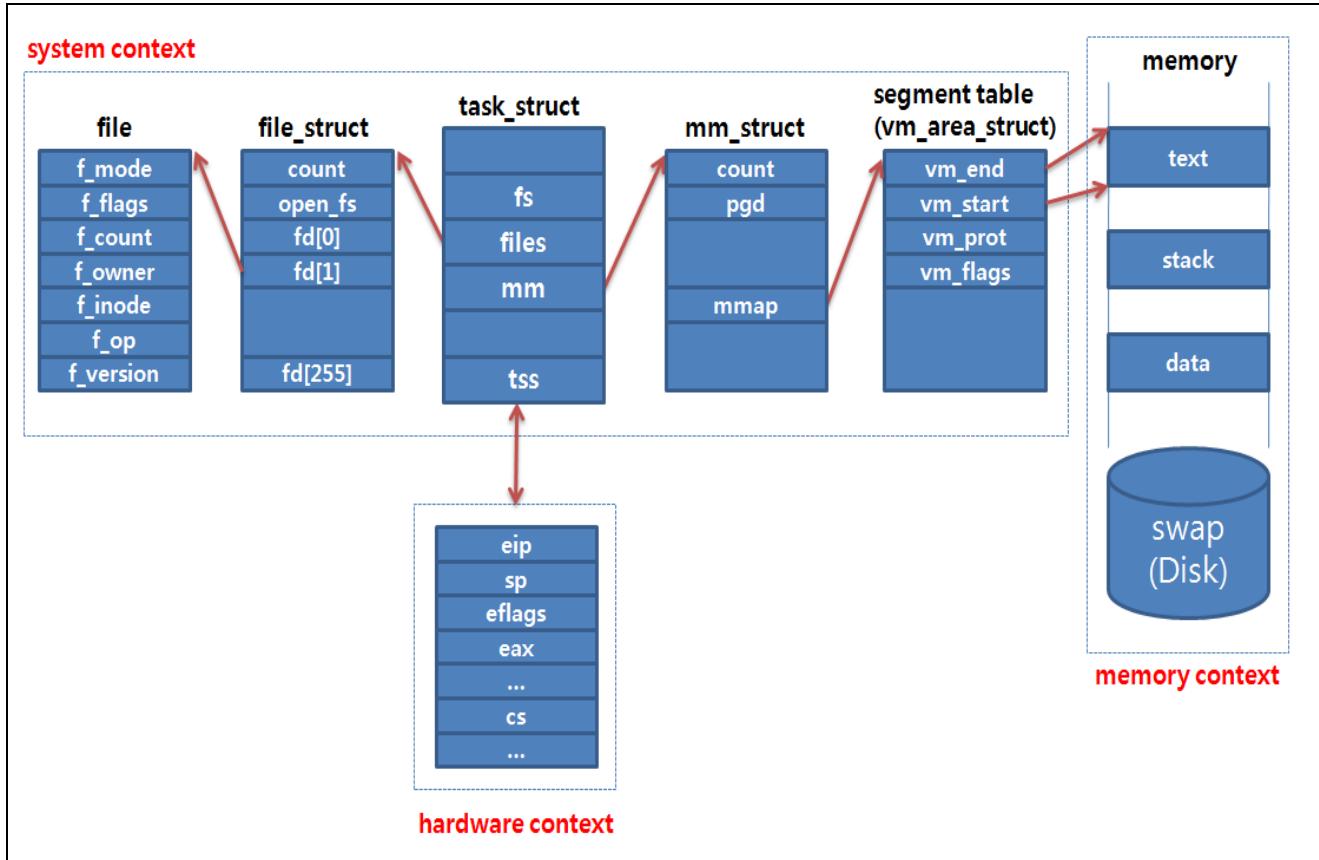
종 목	문 제
컴퓨터시스템응용기술사	4교시 2번

운영체제에서 문맥(context)의 내용, 문맥교환(context switch)의 과정 및 문맥교환이 발생하는 조건을 설명하시오.

도메인/토픽	운영체제 > 문맥교환(Context Switching)
키워드	System context, Memory context, Hardware context Kernel mode, User mode, PCB, Context Save, Context Restore Dispatch, Time run out, Block, Time slice, I/O, System call
풀이 가이드	1. 프로세스 자원과 제어 흐름의 집합, 문맥의 내용 가. 문맥의 유형 -- 시스템, 메모리, 하드웨어 문맥 나. 문맥의 내용 -- Linux의 시스템 문맥 기준 2. 문맥 교환의 과정 가. 문맥 교환의 과정 -- 문맥 교환 과정 도식화 나. 문맥 교환의 과정 설명 -- 커널모드 전환 → Context Save → Context Restore → 사용자모드 전환 3. 문맥 교환이 발생하는 조건 가. 문맥 교환이 발생하는 조건 -- 프로세스 상태 전이도로 표현 나. 문맥 교환이 발생하는 조건 설명 -- dispatch, time slice만료, I/O 작업, System call
필수항목 /연관토픽	Process, Thread, Kernel, PCB, Dispatcher
참고문헌	108회 정보관리/컴퓨터시스템응용기술사 기출풀이, 107회 기술사회, 2016. 2. 27 Virtual Memory:Systems, Carnegie Mellon, 2010. 10. 19 Linux Processes, TLEP, 1999. 1 (http://www.tldp.org/LDP/tlk/kernel/processes.html) TASK관리, 블로그, 2012. 6. 21 (http://mintnlatte.tistory.com/400) Process Context, 신라대학교 컴퓨터공학과, 2008 (slideplayer.com/slide/4842723/)
Advice	Fact를 묻는 문제이기 때문에 문제에 집중한 목차 구성에 기반하여 상세한 내용 작성 시, 고득점이 예상됩니다. 프로세스 상태 전이 및 문맥 교환 관련 문제는 지속적으로 출제 중이므로 반드시 학습이 필요합니다.
작성자	108회 컴퓨터시스템응용기술사 / 김인수 (diun@paran.com)

1. 프로세스 자원과 제어 흐름의 집합, 문맥(Context)의 내용

가. 문맥(Context)의 유형



- 문맥(Context)이란, 여러 프로세스가 함께 수행되는 시분할 시스템 환경에서 특정 프로세스의 처리를 위해 할당 된 자원이 일시적으로 반환되어 다른 선점 프로세스를 처리하고 재 할당 되었을 경우(이를 문맥교환이라 함), 기존 수행 중이었던 작업 시점부터 재현하기 위한 정보를 지칭
- 문맥교환(Context Switching) 발생 시, 이전 프로세스 상태를 재현할 수 있도록 지원하는 정보

시스템 문맥	(System Context) - 커널에 할당된 자료구조 (<code>task_struct</code> , 파일 디스크립터, 세그먼트, 페이지 테이블)
메모리 문맥	(Memory Context) - Text, Data, Stack, Heap 영역, swap 공간
하드웨어문맥	(Hardware Context) - 문맥 교환 시, 현재 실행 위치에 대한 정보를 유지

- 프로세스의 문맥(Context)을 크게 세가지로 분류하면, [시스템 문맥], [메모리 문맥], [하드웨어 문맥]으로 구분 가능

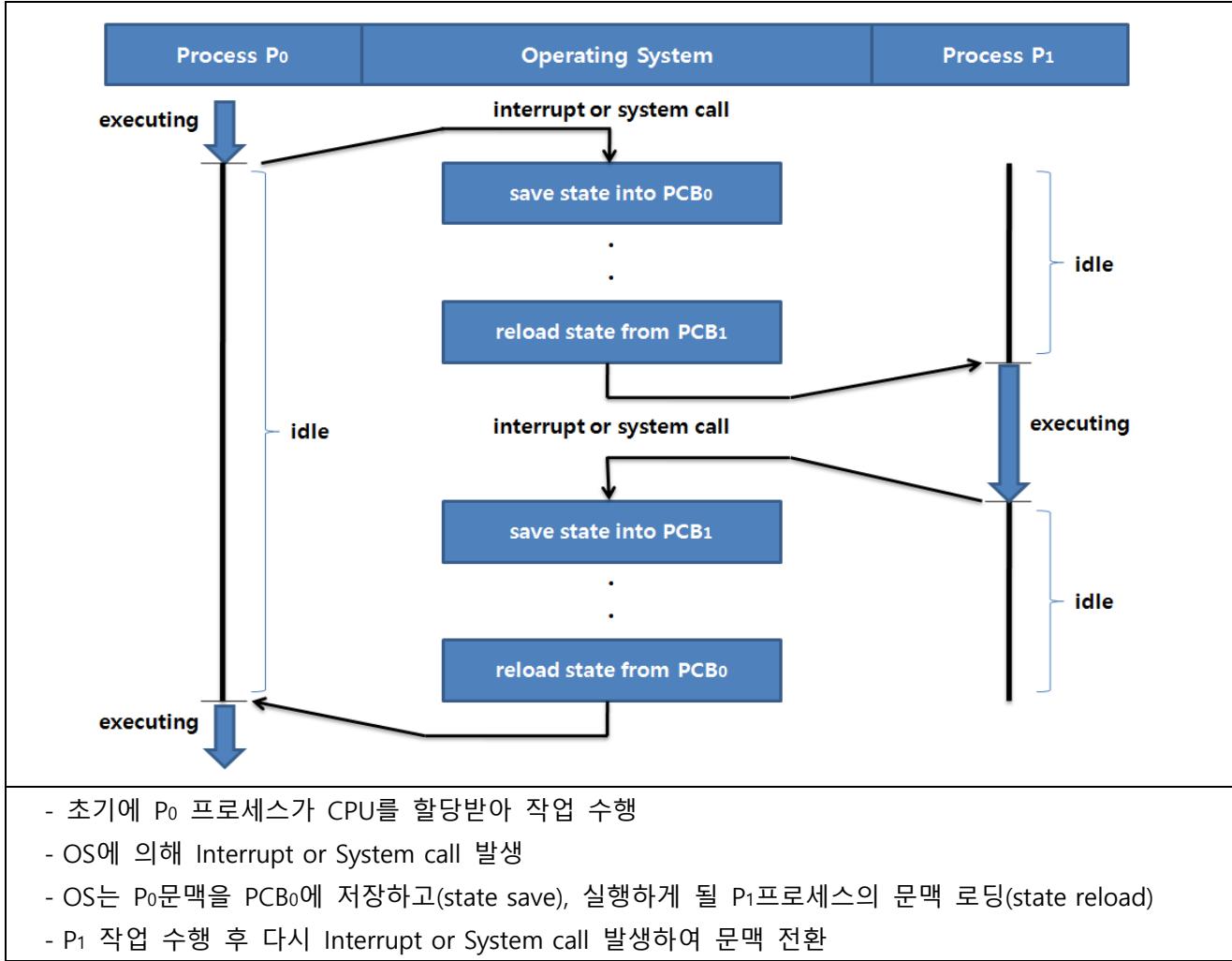
나. 문맥(Context)의 내용 – 시스템 문맥 기준 (/include/linux/sched.h)

구분	설명	
Task identification	- 태스크를 인식하기 위한 변수	
	pid	- 태스크 ID
	tgid	- 태스크가 속해있는 쓰레드 그룹 ID
	사용 접근제어필드	- uid(사용자id), euid(유효 사용자id), suid(저장된 사용자id), fsuid(파일시스템 사용자id)
	그룹 접근제어필드	- gid, egid, sgid, fsgid
State	- 생성에서 소멸까지 상태 관리를 위한 변수	
	- TASK_RUNNING(0), INTERRUPTIBLE(1), UNINTERRUPTIBLE(2), STOPPED(4), TRACED(8), EXIT_ZOMBIE(16), DEAD(32)	
Task relationship	- 태스크는 생성되면서 가족 관계를 가짐	
	real_parent	- 현재 태스크를 생성한 부모 태스크의 task_struct 구조체
	parent	- 현재 부모 태스크의 task_struct 구조체
	children / sibling	- 자식/형제를 리스트로 연결한 뒤, 리스트의 헤드를 저장
Scheduling information	- 태스크에게 비동기적인 사건의 발생을 알리는 메커니즘	
	- signal, sighand, blocked, pending	
Memory information	- 명령어와 데이터를 text, data, stack, heap 공간 등에 저장.	
	- task_struct 공간에 대한 위치, 크기, 접근제어 정보 등을 관리하는 변수 존재	
	- mm_struct	
File information	files	- 태스크가 오픈한 파일 접근 (files_struct 구조체)
	fs	- 루트디렉토리의 inode와 현재 디렉토리의 inode 접근(fs_struct)
Thread structure	- 문맥 교환을 수행할 때 태스크가 현재 어디까지 실행되었는지 기억하는 공간	
	- TSS (Task State Segment)	
Time information	- 태스크의 시간 정보를 위한 변수	
	start_time	- 태스크가 시작된 시간
	cpu_time	- 사용한 cpu시간
	run_delay	- 대기 시간
Format	- 리눅스는 Linux exec 도메인 뿐 아니라 BSD, SVR4 exec 도메인도 지원	
	personality	- 커널에서 컴파일된 프로그램도 재컴파일 없이 수행 위한 변수
	binfmt	- 다양한 binary format을 지원 (a.out, elf, java, shell script)
Resource limits	- 태스크가 사용할 수 있는 자원의 한계. 배열로 구현	
	rlim_max	- 최대 허용 자원 수
	rlim_cur	- 현재 설정된 허용 자원 수

- 프로세스 실행이 전환될 때에 문맥 교환(Context Switching) 발생

2. 문맥 교환(Context Switching)의 과정

가. 문맥 교환(Context Switching)의 과정 도식



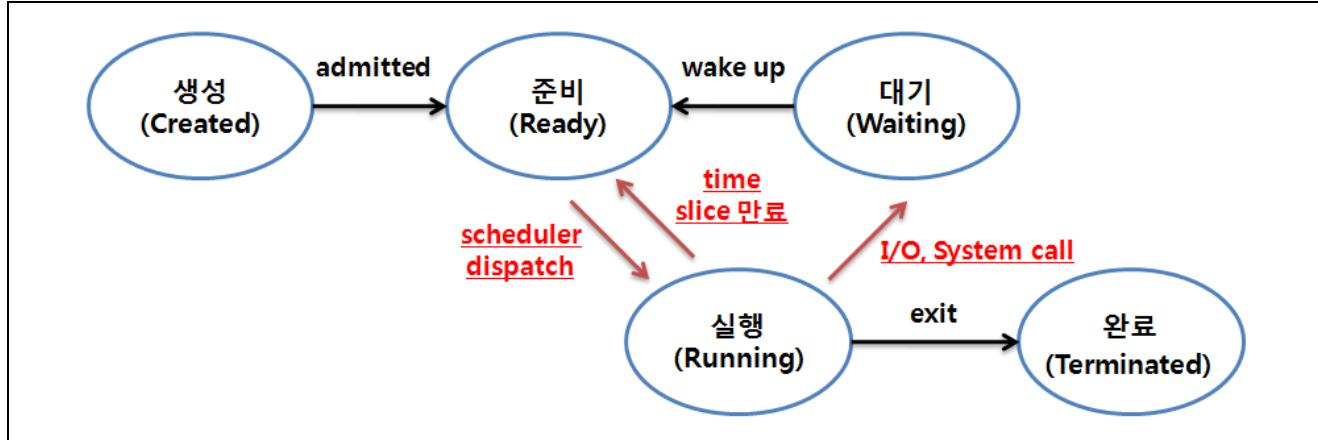
나. 문맥 교환(Context Switching)의 과정 설명

과정	내용
1. 문맥교환 요구	- 인터럽트나 시스템 호출에 의한 문맥 교환 요구
2. 커널 모드 전환	- 사용자 모드 → 커널 모드
3. Context Save	- 기존 프로세스의 현재 상태 정보를 PCB에 저장
4. Context Restore	- 다음에 실행할 프로세스의 상태 정보를 PCB에서 복원
5. 사용자 모드 전환	- 커널 모드 → 사용자 모드

- 대상 프로세스의 정보를 PCB 통해 저장/복원하는 과정 통해 문맥 교환 수행

3. 문맥 교환(Context Switching)이 발생하는 조건

가. 문맥 교환(Context Switching)이 발생하는 조건 – 프로세스 상태 전이도



- Ready → Running, Running → Ready, Running → Waiting 전환 시에 문맥 교환 발생

나. 문맥 교환(Context Switching)이 발생하는 조건 설명

발생 조건	발생 조건 설명	상태 전이 과정
scheduler dispatch	- Ready Queue의 첫 번째 프로세스(가장 높은 우선순위)는 CPU를 할당 받고 실행 상태로 전이	- Ready → Running
time slice만료	- 프로세스가 CPU 사용 가능한 할당 시간 초과 - 시간이 초과되면 PCB저장, CPU 양도 후 준비상태로 전이	- Running → Ready
I/O 작업	- 실행 중인 프로세스가 I/O로 인해 I/O 전용 프로세스에게 CPU 양도 후 대기상태로 전이 - I/O 처리는 CPU가 아니라 I/O 전용 프로세스가 담당하기 때문에 발생	- Running → Waiting
System call	- 실행될 준비가 되어 있지 않은 서비스를 호출할 경우 대기상태로 전이	- Running → Waiting

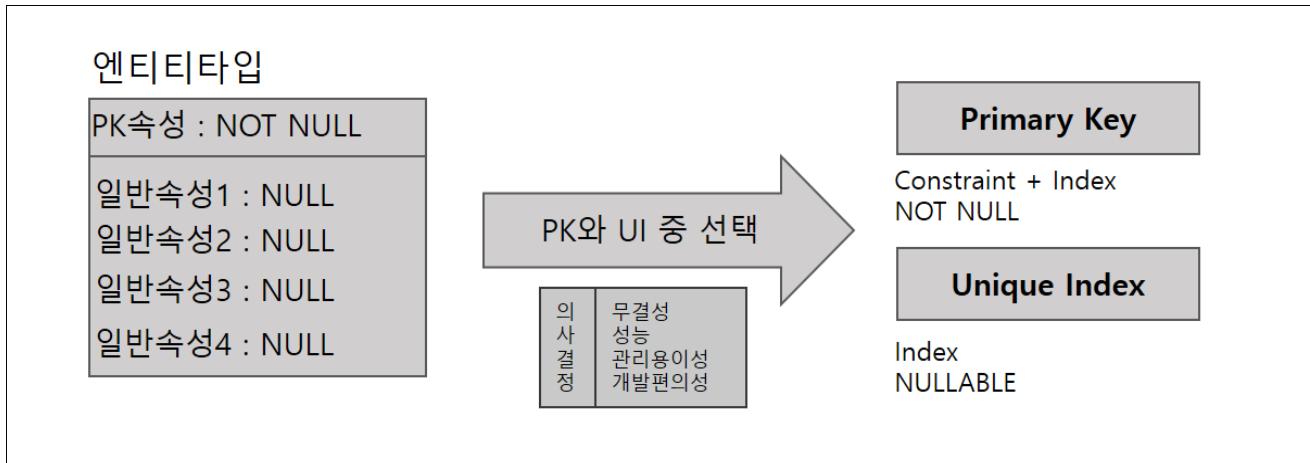
- 문맥 교환 시, 문맥 교환에 의한 오버헤드 발생
- 오버헤드를 해결하기 위해 다중 프로그래밍의 정도를 낮추어 문맥 교환 발생 최소화 및 Thread 이용한 문맥 교환을 수행하여 데이터 전환 최소화

종 목	문 제
컴퓨터시스템응용기술사	4교시 3번

DB 설계에 있어서 Primary Key와 Unique Index의 특성을 비교하고, Primary Key 사용 시의 장점에 대하여 설명하시오.

도메인/토픽	DB > Key DB > Index
키워드	Primary Key - 유일성, 최소성, NOT NULL, 대표성, 테이블 당 개수 1개 Unique Index - 유일성, NULLABLE, 테이블당 복수개 가능 PK장점 - 역공학 시 관계 도출 가능, 참조무결성,
풀이 가이드	1. Primary Key와 Unique Index 를 출제한 출제자의 의도를 반영한 개요 작성 2. Primary Key 와 Unique Index 특성 비교 3. Primary Key 사용 시의 장점 4. PK, UI의 사용 시 고려사항
필수항목 /연관토픽	Primary Key, Unique Index, 제약조건, 참조무결성 Primary Key와 Unique Index 논쟁 정보관리105회 2교시 3번, 정보관리108회 3교시 4번 기출문제
참고문헌	아는 만큼 보이는 데이터베이스 설계와 구축, 한빛미디어(이춘식 저), 2008. 7
Advice	DB 도메인의 기본 토픽으로, 지문에 언급된 각 개념은 이해하고 있더라도, 각각의 특성과, 장단점, 실무자 관점에서 맥을 짚어내는 것이 어려울 수 있는 문제입니다. 또한 분량에 신경 쓰시다가 물어본 특성, 장점 등의 논점이 아닌 아는 것을 길게 쓰는 우를 범하기 쉽습니다. 이 부분에 주의하고 최대한 문제에 집중하시어 문제에서 물어본 특성, 장점을 학습과정을 통한 지식을 연계하여 기술하였다면 고득점 획득이 가능합니다.
작성자	108회 컴퓨터시스템응용기술사 / 문영호 (hamelmoon@gmail.com)

1. 데이터베이스 설계의 Dispute Point, PK 와 UI



- DB설계 실무에 있어, PK와 UI를 올바르게 선택하는 것은, 차후 시스템의 개발, 유지보수, 성능 관리 등에 큰 영향을 주는 중요한 설계 포인트.
- SAP ERP, Oracle ERP 등에서는 각 국가 및 기업의 다양한 비즈니스 환경에서도 유연하게 적용할 수 있도록 OSMU기반의 ERP에 Unique Index를 사용
- 일반적인 개발환경에서 설계 시, "무결성", "성능", "관리 용이성", "개발 편의성"과 같은 다양한 측면을 고려하여 UI(Unique Index)보다는 PK(Primary Key)를 선택하는 하는 것이 유리.

2. Primary Key 와 Unique Index의 사용 및 특성 비교

가. Primary Key 와 Unique Index 의 사용

구분	PK(Primary Key)	UI(Unique Index)																																																				
설명	<ul style="list-style-type: none"> - 관계에 저장된 레코드를 고유하게 식별하는 후보 키(속성 또는 속성의 집합) 가운데 기본 값으로 지정해 놓은 주 키 	<ul style="list-style-type: none"> - 컬럼 내 중복 데이터 비 허용 인덱스 - 복수 컬럼의 경우, 컬럼들의 합이 유일 레코드로 생성 가능토록 하는 인덱스 																																																				
적용 예시	<div style="display: flex; align-items: center;"> <div style="border: 1px solid blue; padding: 10px; margin-right: 20px;"> <p>팀 테이블의 TEAM_NAME 컬럼 인덱스 생성</p> <p>인덱스를 활용한 SQL</p> <pre>SELECT TEAM_NAME, TEAM_NAME FROM TEAM WHERE TEAM_NAME IN ('구매1팀','영업1팀')</pre> </div> <div style="flex-grow: 1; position: relative;"> <div style="position: absolute; top: -10px; left: 0; background-color: #e0f2e0; width: 100%; height: 10px;"></div> <div style="display: flex; align-items: center; gap: 20px;"> <div style="border: 1px solid lightgray; padding: 5px; background-color: #f0f8ff;"> <p>팀 이름 (TEAM_NAME) 인덱스</p> <table border="1" style="width: 100%; border-collapse: collapse;"> <thead> <tr> <th>TEAM_NAME</th> <th colspan="3">ROW ID</th> </tr> <tr> <th></th> <th>BLOCK#</th> <th>LINE#</th> <th>FILE#</th> </tr> </thead> <tbody> <tr> <td>IT팀</td> <td>000000A</td> <td>.0001</td> <td>.0001</td> </tr> <tr> <td>구매1팀</td> <td>000000A</td> <td>.0004</td> <td>.0001</td> </tr> <tr> <td>구매2팀</td> <td>000000A</td> <td>.0005</td> <td>.0001</td> </tr> <tr> <td>영업1팀</td> <td>000000A</td> <td>.0002</td> <td>.0001</td> </tr> <tr> <td>영업2팀</td> <td>000000A</td> <td>.0003</td> <td>.0001</td> </tr> </tbody> </table> </div> <div style="border: 1px solid lightgray; padding: 5px; background-color: #f0f8ff;"> <p>팀 (TEAM) 테이블</p> <table border="1" style="width: 100%; border-collapse: collapse;"> <thead> <tr> <th>TEAM_NO</th> <th>TEAM_NAME</th> <th>MNG_EMP_NO</th> <th>COST_MNG_EMP_NO</th> </tr> </thead> <tbody> <tr> <td>10001</td> <td>IT팀</td> <td>1</td> <td>7</td> </tr> <tr> <td>10002</td> <td>영업1팀</td> <td>2</td> <td>8</td> </tr> <tr> <td>10003</td> <td>영업2팀</td> <td>3</td> <td>8</td> </tr> <tr> <td>10004</td> <td>구매1팀</td> <td>4</td> <td>8</td> </tr> <tr> <td>10005</td> <td>구매2팀</td> <td>5</td> <td>8</td> </tr> </tbody> </table> </div> </div> </div> </div>	TEAM_NAME	ROW ID				BLOCK#	LINE#	FILE#	IT팀	000000A	.0001	.0001	구매1팀	000000A	.0004	.0001	구매2팀	000000A	.0005	.0001	영업1팀	000000A	.0002	.0001	영업2팀	000000A	.0003	.0001	TEAM_NO	TEAM_NAME	MNG_EMP_NO	COST_MNG_EMP_NO	10001	IT팀	1	7	10002	영업1팀	2	8	10003	영업2팀	3	8	10004	구매1팀	4	8	10005	구매2팀	5	8	<p>PK</p> <p>UI</p>
TEAM_NAME	ROW ID																																																					
	BLOCK#	LINE#	FILE#																																																			
IT팀	000000A	.0001	.0001																																																			
구매1팀	000000A	.0004	.0001																																																			
구매2팀	000000A	.0005	.0001																																																			
영업1팀	000000A	.0002	.0001																																																			
영업2팀	000000A	.0003	.0001																																																			
TEAM_NO	TEAM_NAME	MNG_EMP_NO	COST_MNG_EMP_NO																																																			
10001	IT팀	1	7																																																			
10002	영업1팀	2	8																																																			
10003	영업2팀	3	8																																																			
10004	구매1팀	4	8																																																			
10005	구매2팀	5	8																																																			
생성	ALTER TABLE TEAM	CREATE UNIQUE CLUSTERED INDEX																																																				

SQL 구문	ADD CONSTRAINT PK_TEAM PRIMARY KEY CLUSTERED (TEAM_NO)	UIDX_TEAM_NAME ON TEAM (TEAM_NAME)
--------	--	------------------------------------

- Primary Index(PK)를 정의하면 물리적으로 Uniqueness를 보장하기 위해 Unique Index(UI)가 만들 어지기 때문에 성능향상의 효과를 얻을 수 있음.
- 일반적으로 PK를 기준으로 CRUD 작업이 수행되기 때문에, PK 정의 시, WHERE 조건 절에 PK가 SARGs(Search Arguments)로 사용된 쿼리의 성능은 향상되나, PK가 없으면 Table Full Scan 발생
- OLAP Tool 등에서 ADHOC 쿼리 처리 시, 성능개선을 위한 용도로 UI를 사용

나. Primary Key 와 Unique Index 의 특성 비교

항목	PRIMARY KEY	UNIQUE INDEX
목적	- Constraint + Index	- Index
공통점	- 유일성 보장	- 유일성 보장
참조무결성	- PK/FK에 의해 지정 가능	- 지정 불가능
데이터 무결성 처리 레벨	- DBMS Level	- Application Level (Procedure , Source code)
테이블당 개수	- 1개만 가능	- 여러 개 가능
인덱스 생성 역공학 적용 시	- Unique Index 생성	- Unique Index 생성
NULL 허용	- PK 인식 가능 - 허용 안됨	- PK인식 불가능 - 허용됨
포함 조건	- Not Null Constraints - Unique index	- Unique index

- Unique Index만 사용하여 PK를 정상적으로 생성하지 않으면, 데이터 모델에 대한 파악이 어려움.
- 중요하고 정확한 데이터를 다루는 업무에서 Unique Index 사용 시, 무결성 유지가 관건.

3. Primary Key 사용 시의 장점

가. Primary Key 없이 Unique Index 사용 시 장/단점

항목	장점	단점
무결성	- 없음	- 데이터 무결성 취약
성능	- PK/FK를 이용하지 않으므로 성능향상 기대	- 테이블 관리비용 증가
관리 용이성	- PK/FK가 존재하지 않아 DBA의 데이터 베이스 관리가 용이 (다수 Join에 따른 성능저하 없음)	- 데이터모델과 테이블간의 관계가 불일치하여 관리 어려움 - Unique Index가 복수개 생성될 경우, → PK가 무엇인지 구분 불가

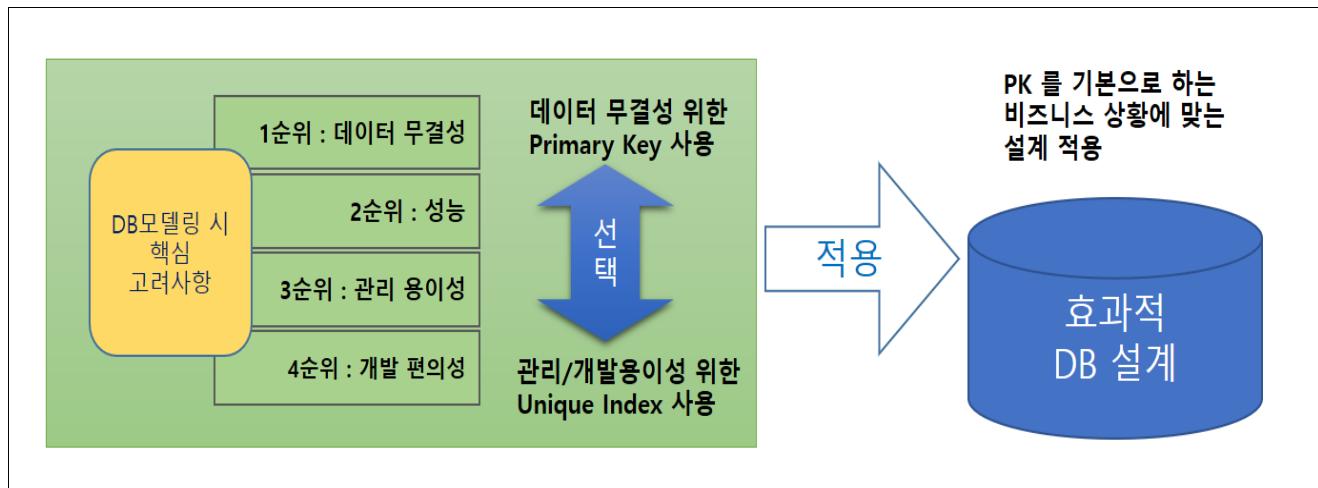
개발 편의성	<ul style="list-style-type: none"> - 개발 시점에 데이터 제약이 없으므로 개발이 용이 - 역공학 모델링 시, 관계도출이 어려움
<ul style="list-style-type: none"> - Unique Index 만을 사용할 경우 성능, 관리 면에서는 어느 정도 효율적일지 모르나, Unique Index의 무분별한 사용으로 데이터 정합성이 깨질 우려가 있으므로 PK를 사용을 권장. 	
나. Primary Key 사용 시의 장점	

나. Primary Key 사용 시의 장점

항목	장점	비고
무결성	<ul style="list-style-type: none"> - PK/FK 관계 설정을 통한 참조무결성 제약조건 지정함으로써 무결성 오류를 방지 	<ul style="list-style-type: none"> - 데이터 정합성 증가
	<ul style="list-style-type: none"> - FK를 설정하지 않는다면, UI(Unique Index)만을 사용하는 것 대비, 성능 저하가 없음. 	<ul style="list-style-type: none"> - FK인덱스 활용
관리 용이성	<ul style="list-style-type: none"> - 테이블당 한개의 PK만 허용되므로 역공학 적용 데이터 모델 생성 시, PK가 구분 가능. 	<ul style="list-style-type: none"> - ERD 툴 통한 역공학 모델링
개발 편의성	<ul style="list-style-type: none"> - PK 사용을 통해, 무결성을 위한 Data Validation 부담 완화 	<ul style="list-style-type: none"> - 데이터 정합성 증가

- 일반적 개발 상황에서 PK를 사용하는 것이 많은 장점이 있으며, Unique Index가 PK에 비해 장점이 있다면 개발과 유지보수에 약간의 편리함을 제공 한다는 것이다

4. PK, UI의 사용 시 고려사항



- 실무 적용 시, 기본적으로 PK를 선택하여 사용하는 것을 기본 함
- 글로벌 패키지 프로그램 등 비즈니스 모델의 차이가 있지만 유연하게 적용 필요한 경우
- DBMS의 Procedure, 소스코드 등에 무결성을 유지시키기 위한 로직을 포함하여 제약조건을 만족
- PK와 FK에 인덱스를 적절히 활용하여, 성능에 대한 우려를 극복

종목	문제
컴퓨터시스템응용기술사	4교시4번

통신망(Communication Network) 장치들의 물리적 구성 형태 및 각각의 특징에 대하여 설명하시오.

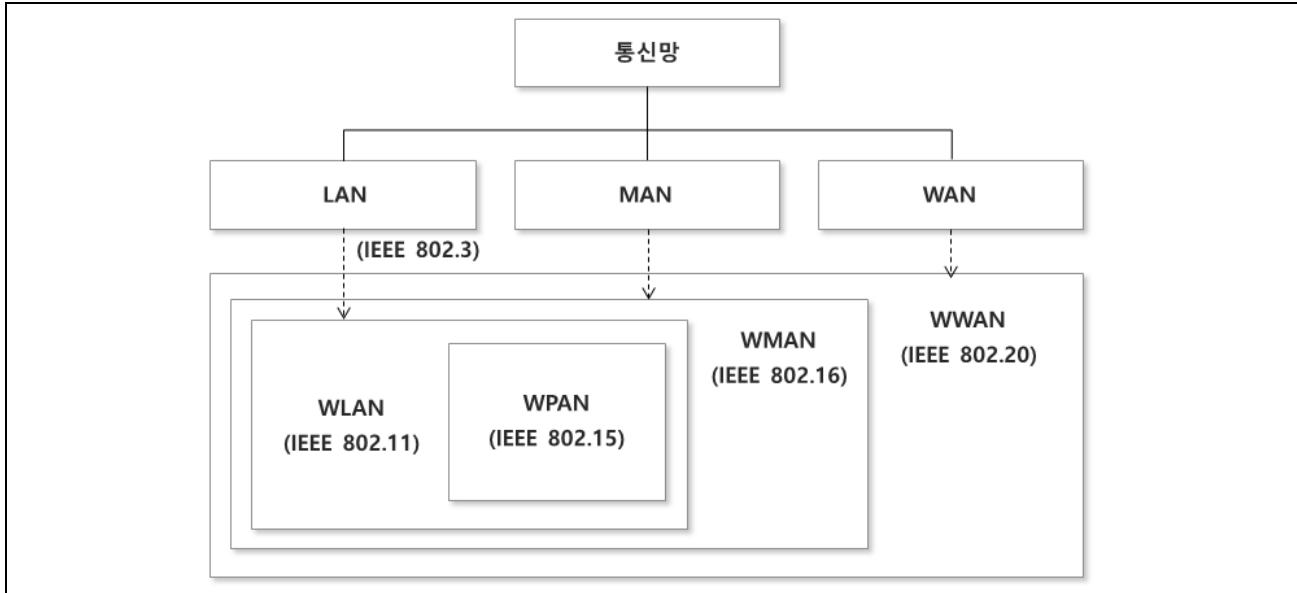
도메인/토픽	통신>통신망(Communication Network)
키워드	LAN/WAN, Hub/Repeater/Bridge/Switch/Router, 성형/버스형/트리형/링형/그물형
풀이가이드	<ol style="list-style-type: none">통신망의 개요통신망을 구성하는 장치통신망 구성 장치들의 물리적 구성 형태환경에 따른 물리적 구성 적용 제안
필수항목 /연관토픽	데이터통신과 네트워크의 차이점 이해 OSI 7계층과 토폴로지 간의 연관관계 이해
참고문헌	데이터통신과 네트워킹, Mc Graw Hill (Forouzan 저), 2007. 8 정보통신설비설계 공사실무, 성악당(채해수 저), 2015. 1
Advice	통신망을 구성하는 물리적 장치에 대하여 묻는 것이 아니고, 장치의 물리적 구성을 묻는 것이기 때문에, 문제에 대한 집중이 필요함
작성자	108회컴퓨터시스템응용기술사 / 문정기 (jvioonpe@naver.com)

1. 통신망의 개요

가. 통신망(Communication Network)의 정의

- 송수신자간 통신 경로를 구성하기 위해 단말, 회선, 전송장비, 교환기, 프로토콜 등이 유기적으로 결합하여 구성한 통신장치들의 집합

나. 통신망의 유형



- 통신망은 거리에 따라 LAN(근거리), MAN(지역망), WAN(광역통신망)으로 구분할 수 있음.

2. 통신망을 구성하는 장치

가. OSI 7Layer와 각 계층별 장치

Application		Application	사용자 인터페이스 담당
Presentation		Presentation	데이터 표현, 암호화 담당
Session		Session	응용 프로그램간의 세션을 형성, 어플리케이션 접근 스케줄링 담당
Transport		Transport	TCP와 UDP 담당, 데이터 에러 검출 후, 재전송 담당
Network	Router	Network	경로 결정에 사용할 논리적 주소 제공
Data-Link	Bridge, Switch	Data-Link	MAC Address를 사용하여 매체 접근
Physical	Hub, Repeater	Physical	장비들 사이에서의 비트 이동 담당

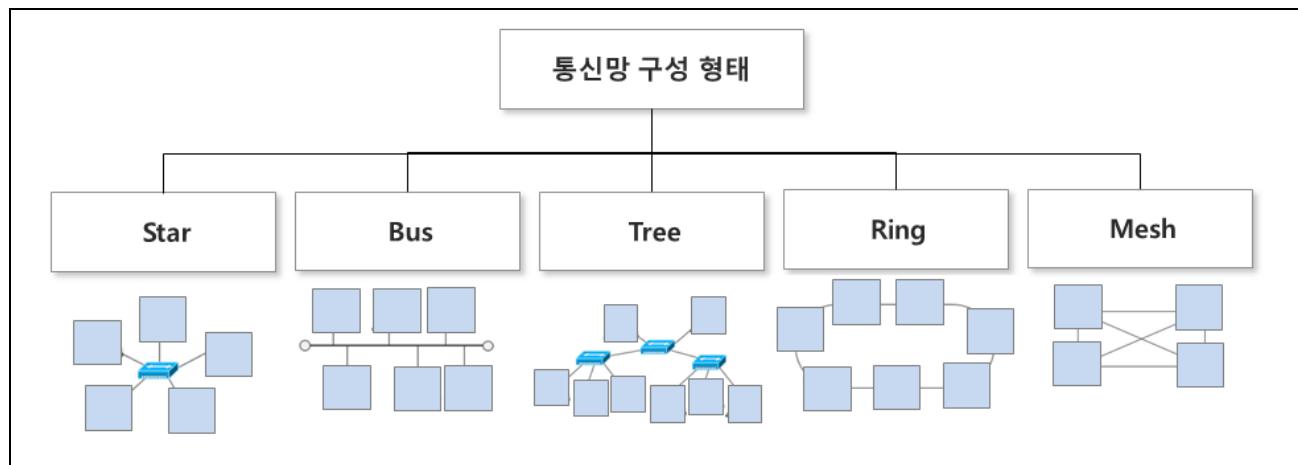
나. 계층별 장치의 특징

장치	설명	OSI 7Layer 계층
Hub	- 여러 개의 포트를 가지고 있으며, 한 포트에서 수신된 신호가 모든 다른 포트에 공유되도록 하는 장치	1계층
Repeater	- 데이터 신호를 수신하여 증폭시킨 후, 다음 구간으로 재전송하는 장치	1계층
Bridge	- 두 개 이상의 LAN을 연결하여 하나의 네트워크로 만들어주는 장치	2계층
Switch	- PC를 연결시켜주고 할당되는 대역폭을 극대화 시켜 주는 장치	2계층
Router	- 서로 다른 구조의 망을 가지는 네트워크 간의 통신을 위해 사용하는 장치	3계층

- 대규모 WAN의 관점에서 볼 때에는 SDH/SONET, MSPP, OXC/OADM 등의 장치들이 포함될 수 있음

3. 통신망 구성 장치들의 물리적 구성 형태

가. 통신망의 물리적 구성 유형



- 사용목적에 따라 확장성, 전송효율, 경제성 등을 감안하여 결정하여야 함.

나. 통신망 구성 형태별 특징

구성형태	특징
성형(Star)	<ul style="list-style-type: none"> - 중앙에 전송제어장치(ex. 허브)가 위치하고 다른 모든 노드를 연결한 형태 - 메인프레임이 전형적인 성형 형태이며, 연결된 각 장치는 직접적 통신이 불가능하고, 전송제어장치를 통해서만 통신이 가능 - 즉 데이터를 전송하기 위해서는 중앙의 전송제어장치로 데이터가 먼저 보내져야만 함 - 설치 비용이 저렴하고 유지보수가 용이하다는 장점이 있으나, 제어장치에 문제가 생길 경우, 전체가 그 영향을 받으며, 통신량이 많을 시에는 전송이 지연되는 단점이 있음
버스형(Bus)	<ul style="list-style-type: none"> - 모든 노드와 주변장치가 일자형 케이블에 연결된 형태 - 하나의 노드가 전송을 하면 그것이 브로드캐스트되어 다른 모든 노드가 데

	<p>이터를 수신</p> <ul style="list-style-type: none"> - 즉, 모든 LAN상의 사용자에게 데이터를 전송하는 1:N 통신을 하여, 수신을 받은 노드들은 수신측 주소를 확인해, 데이터가 자신에게 보내진 것이면 수신하고, 그렇지 않으면 흘려보냄 - 설치가 간단하고 경제적이나, 중앙 케이블에 문제가 생기면 전체가 영향을 받게 됨 - 데이터 전송 시, 이미 다른 노드에서 데이터를 전송하고 있으면 충돌이 발생하고, 이 데이터는 나중에 다시 전송해 주어야 함 - 트래픽이 많아지거나 노드 수가 늘어나면 충돌이 잦아지게 되고 네트워크 성능이 저하된다는 단점이 있음
트리형(Tree)	<ul style="list-style-type: none"> - 성형의 변형으로 중앙 전송제어장치에 모든 노드가 연결되는 것이 아니라, 트리 모양으로 전송제어장치를 두는 형태 - 계층적 네트워크에 적합하며, 성형과 같이 제어가 간단하고 관리/확장이 쉬우나, 성형이 가지는 단점인 중앙 전송제어장치 장애에 대한 문제가 동일하게 발생
링형(Ring)	<ul style="list-style-type: none"> - 모든 노드가 하나의 링에 순차적으로 연결되는 형태 - 실제로 링과 같은 모양을 가진다기 보다는 처음과 끝이 연결되는 링과 같은 모습을 가져 링형이라고 불림 - 구조가 단순하고 장애 발생 시의 복구가 빠르나, 링을 제어하는 절차가 복잡하고 노드의 추가가 어려운 단점이 있음 - 만약 하나의 노드에 문제가 생기면 전체 네트워크가 사용불가하게 되며, 이 문제를 개선하기 위해 이중링을 사용하거나 스위치를 사용할 수 있음
그물형(Mesh)	<ul style="list-style-type: none"> - 모든 노드가 Point-to-Point로 연결되는 형태로, 모든 노드가 연결되기 위해서는 $n(n-1)/2$개의 회선이 필요 - 각 노드는 자신을 제외한 $n-1$개의 입출력 포트를 가지고 있어야 하며, 많은 회선을 필요로 하기 때문에 비용이 높으나 보다 높은 신뢰성을 제공함 - 데이터의 전송을 보장하고, 원하는 수신자만 받을 수 있기 때문에 보안 측면에서 우수함. 또한 하나의 노드나 회선에 문제가 생기더라도 전체적인 네트워크에 영향이 적으며, 공유로 발생하는 통신량 문제도 해결함 - 장애 발생 시, 결함의 식별이 쉽고 분리가 용이하나, 네트워크의 재구성이 어렵고, 무엇보다 비싼 비용이 가장 큰 단점임

- 실제 네트워크 구성 시에는 목적에 따라 위의 형태들을 혼합하여 사용되는 경우가 많음.

4. 환경에 따른 물리적 구성 적용 제안

- LAN 또는 중소규모 WAN을 구성하고자 할 때에는 버스형 or 성형이 주로 사용됨.
- WAN을 구성하고자 할 때에는 링형이 주로 사용되며, RPR은 링형의 한 사례임.

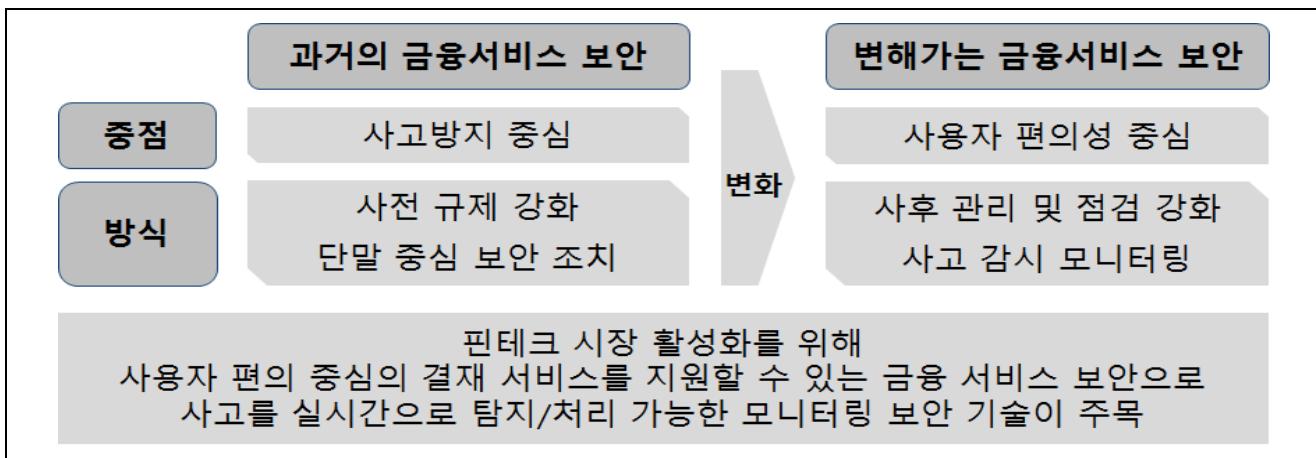
항목	성형	버스형	트리형	링형	그물형
설치비용/노력	초기 비용 큼	경제적	초기비용 큼	설치간단 (제어절차가 복잡)	설치가 어려움 (회선, HW)
재구성	용이	어려움	제어구조에 따라 다름	용이 (구조단순)	어려움
장비추가	용이	용이	용이	어려움 (링 절단)	규모에 따라 다름
케이블수/비용	케이블은 적게 사용, 중앙까지 연결필요	비교적 적게 필요 (경제적)	2~n차 제어 장치에 따라 다름	성형보다 저렴	많이 필요

종 목	문 제
컴퓨터시스템응용기술사	4교시 5번

FDS(Fraud Detection System)의 개념, 구성요소 및 기능을 설명하시오

도메인/토픽	정보보안 > FDS(Fraud Detection System)
키워드	정보수집-분석을 통한 탐지 및 대응, 이용자정보, 사고유형정보, 오용탐지/이상탐지,
풀이 가이드	1. 등장배경 및 현황 설명 등의 간결한 소개 (2~3번의 설명을 위한 사전 소개) <u>2. FDS의 개념</u> <u>3. FDS의 구성요소</u> <u>4. FDS의 기능</u> 5. FDS 도입 시의 고려사항 [문항에서 물어보는 항목인 2~4번에 대해서는 반드시 작성을 하셔야 합니다]
필수항목 /연관토픽	핀테크, FIDO, 비대면금융거래, FDS, 금융거래 정보를 이용한 빅데이터 관련 서비스 및 빅데이터 보안
참고문헌	이상금융거래 탐지시스템 기술가이드, 금융보안원, 2014. 8 금융권 이상 금융 탐지 시스템(FDS) 고도화 1.0, 금융감독원, 2014. 12 ITU-U x.sap-7(x1157) "Technical capabilities of fraud detection and response for service with high assurance level requirements"
Advice	핀테크로 변화해가는 온라인금융결제 시스템 보안을 위한 기술들이 출제됨에 따라 이후에는 각각의 기술이 아닌, 종합적인 보안수립 묻는 문제가 나올 가능성이 있으므로 국내외 핀테크 서비스 동향 및 서비스별 보안취약점과 취약점 보완을 위한 기술을 종합적으로 연계하여 대비할 필요가 있습니다.
작성자	108회 컴퓨터시스템응용기술사 / 심준영 (losthk@naver.com)

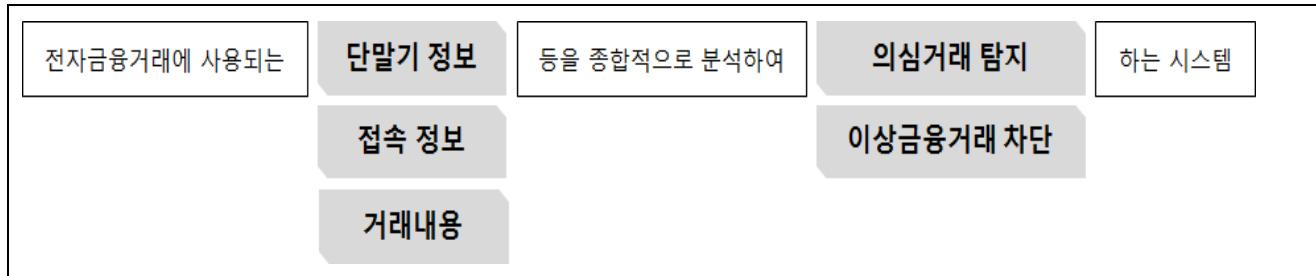
1. 사용자 편의성 중심의 결재시장 활성화 위한 금융보안 서비스의 변화



- 간편결재, 인터넷은행과 같은 사용자 편의성 중심의 금융서비스인 fintech 시장의 활성화를 위해 결재간 수행되는 복잡한 금융보안서비스의 변화가 요구되며, 이에 따라 이상금융거래를 실시간으로 판단/처리 가능한 FDS가 주목

2. FDS(Fraud Detection System)의 개념

가. 이용자의 간편하고 안전한 금융거래 지원을 위한 보안시스템, FDS의 정의



- 안전한 전자금융거래를 위해, 전자 금융거래에 사용되는 이용자 정보 및 이상금융거래 사례 등을 종합적으로 분석하여 의심거래를 탐지하고 탐지되는 이상금융거래를 차단하는 시스템

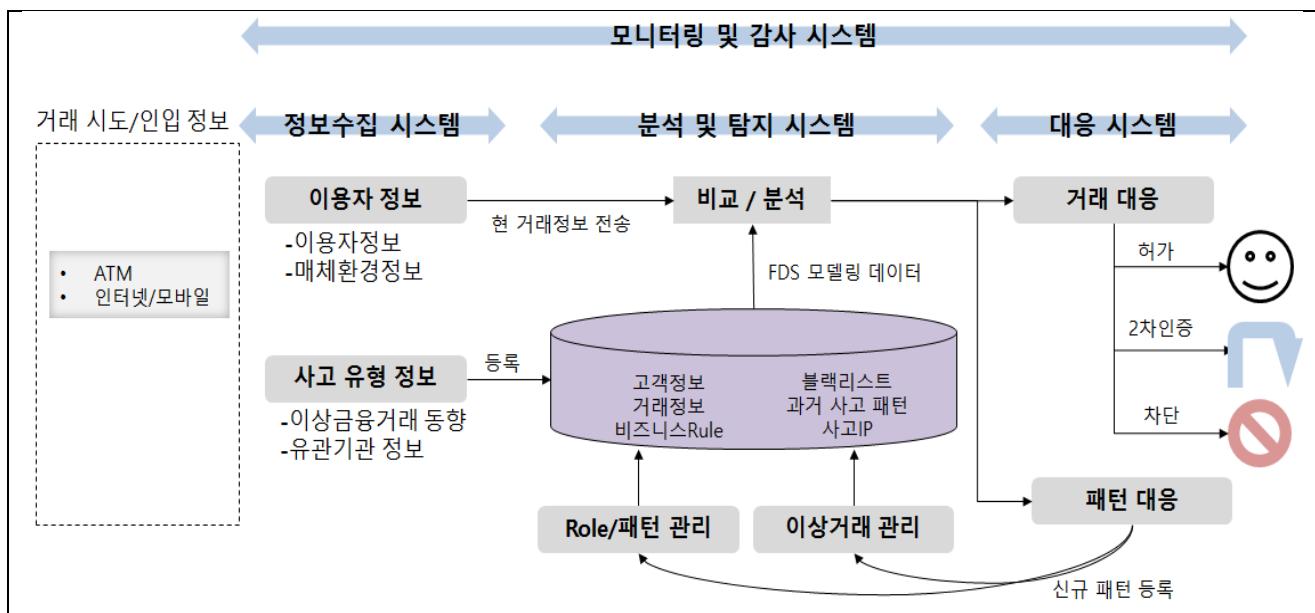
나. FDS구축 지원 위한 국내외 주요 활동

美, FFIEC의 '인터넷뱅킹 인증 가이드라인'	국내 금융보안원의 FDS가이드
2011년 6월, 미국 연방 금융기관 검사 협의회(FFIEC)에서 발표한 '인터넷뱅킹 인증 가이드라인'에서는 금융회사가 전자금융 부정거래를 적시에 효과적으로 대응할 수 있도록 부정거래 탐지 및 모니터링 시스템(FDS)의 구축을 권고	금융보안원은 IT융합금융서비스 활성화를 위한 보안성검토, 보안가이드 개발 등과 같은 지원 활동을 수행하고 있으며, 이러한 활동 일환으로 "이상금융거래탐지시스템(FDS) 기술가이드"를 배포하여 금융권에서의 FDS구축 및 운영을 지원

- 이 외에도 ITU-U x.sap7(Technical capabilities of fraud detection and response for service with high assurance level requirements)과 같은 FDS관련 국제 표준 제정.

3. FDS의 구성

가. FDS의 구성도



- FDS의 구성요소는 일반적으로 '정보수집시스템', '분석/탐지시스템', '대응시스템', '모니터링 및 감사 시스템'으로 분류되나, 금융기관의 사용목적과 환경에 따라 조직에 적합하게 구성되어야 함

나. FDS의 구성요소

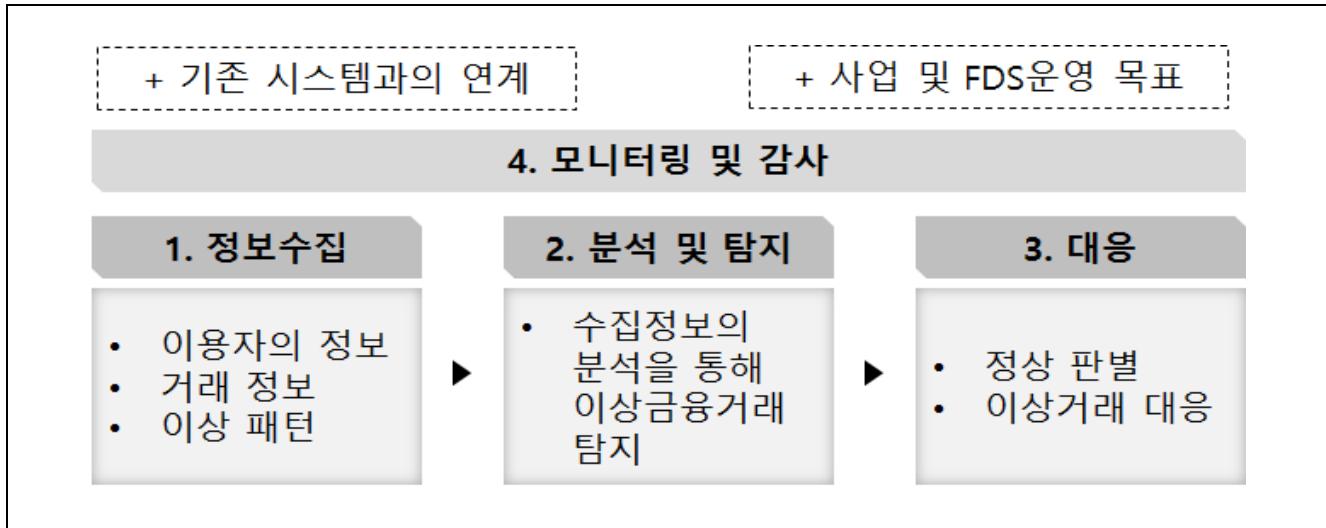
구성요소	세부구성	설명	
정보수집 시스템	이용자 정보	이용자 기본정보	- 이름, 계좌번호, 지정단말 정보
	이용자 매체 환경 정보	디바이스 정보	- 접속된 사용자의 디바이스 정보 (NW환경, 장치유형, MAC 등)
		위치정보	- 접속된 사용자의 위치 정보 (물리적-국가위치/주소) (사이버- IP주소)
	사고유형 정보	국내외 이상금융 거래 동향	- 금융회사, 유관기관 및 기타 단체에서 지속적으로 발생하는 이상금융거래 정보 동향을 지속적으로 검색/수집하여 FDS에 반영
		유관기관 정보	
		악성코드 분석	- 발생된 금융사고에 대한 포렌식분석 모델을 통해 FDS에 반영
분석 및 탐지 시스템	분석시점	실시간 분석	- 거래종료 전에 분석/처리
		준 실시간 분석	- 거래종료 이후, 일정시간 내에 분석/처리
	이상탐지 시스템	오용탐지 모델	- 알려진 사고에 대한 차단목적의 탐지 - 패턴탐지모델, 상태전이 모델 적용 통한 시그니처 기반 / 지식기반 탐지 방식
		이상탐지 모델	- 알려지지 않은 사고에 대한 예방목적의 탐지 - 통계모델, 데이터 마이닝 모델 적용통한 확률 기반/예측기반 탐지 방식
대응 시스템	금융거래 대응시스템	의심/부정거래로 판단되는 경우, 이에 대한 사용자 추가인증 및 차단	

	패턴 대응 시스템	추가패턴 반영	- 정상/비정상 거래에 대한 대응이 완료될 경우, 해당 패턴에 대한 즉각적 반영을 통해 이후 분석/탐지에 활용
		이상금융거래 정보 공유	- 사고로 분류된 이상거래 패턴은 즉각적으로 공유되어 타 금융사에서의 동일사고를 사전에 예방
모니터링 /감사 시스템	모니터링	운영모니터링	- FDS에 대한 시스템 상태 모니터링
		거래모니터링	- 금융거래에 대한 정상/이상탐지 모니터링
	감사	시스템 감사	- 탐지시스템 침해에 대한 감사 기능 수행

- 이용자 및 확정된 금융사고유형 정보를 '수집 및 분석'하여 부정거래에 대한 즉각적인 '대응'을 수행함으로써 <안전한 금융거래 지원> 및 <능동적인 금융사기 패턴의 확보> 가능.

4. FDS의 기능

가. FDS의 기능 구성



- FDS는 기본적으로 '정보수집기능', '분석 및 탐지기능', '거래대응 기능', '모니터링 및 감사' 기능을 포함하고 있으며, 추가적으로 '기존시스템과의 연계기능', '사업 및 운영목표'가 요구됨.

나. FDS의 기능 설명

기능	세부기능	설명
정보수집기능	- 이용자의 금융정보, 거래행위정보 및 알려진 사고유형에 대한 정보 수집	
	이용자 매체정보 수집	- Agent 등을 활용하여 이용자의 금융거래 정보를 수집
	금융거래 유형 정보 수집	- 이용자의 정상적인 거래 패턴 정보 수집 및 모델링
	사고 유형정보 수집	- 국내외 이상금융거래 동향을 통한 사고 유형정보 획득 - 발생한 사고에 대상의 포hen식 통한 사고 유형정보 획득
분석/탐지	- 수집된 정보를 통한 다양한 분석 및 규칙검사를 수행함으로써 이상행위를 탐지	
	이용자 매체정보 활용한 탐지 기능	- 이용자가 사전에 지정한 환경(IP, 국가, 단말)과 상이한 환경에서의 접속 시도 탐지
	이용자 변경정보 활용한 탐지 기능	- 공인인증서 재발급, 고객정보 변경 등의 시도 탐지
	금융거래 정보를 활용한 탐지 기능	- 이용자 금융거래패턴과 다르게 발생하는 거래시도 탐지
대응 (금융거래대응)	차단/통지기능	- 확실시 되는 이상 금융거래에 대해서는 지체없이 차단하고, 사고 내용을 통지하는 기능
		- 사전분석 된 위험모델과 동일한 거래행위 탐지 시, 거래 차단이 적용됨과 동시에 해당 내용을 통지하는 기능
	추가인증 기능	- 신규/의심되는 패턴의 금융거래 발생 시, 2차 인증 수행 → 해당 결과에 따른 거래 승인/차단 적용
	추가 패턴 반영 기능	- 정상/비정상 거래에 대한 대응이 완료될 경우, 해당 패턴에 대한 즉각적 반영을 통해, 이후 분석/탐지에 활용
대응 (패턴대응)	이상금융거래 정보 공유 기능	- 사고로 분류된 이상거래 패턴은 즉각적으로 공유되어 타 금융사에서의 동일 사고를 사전에 예방
	데이터 표준화	- 각 금융사간 패턴정보 공유를 위한 이상거래탐지 데이터의 표준화(IETF의 Sharing Transaction Fraud Data 등)
감시 및 모니터링	모니터링	- 시각화 등을 통한 효율적인 모니터링 감시 (운영 / 거래 / 대응 현황 등 모니터링)
		- 시스템 자율적으로 FDS의 각 구성요소 동작 모니터링
	감사	- 탐지시스템 침해에 대응하기 위한 감사 기능
연계 기능	각 단계별 연동	- 수집-분석탐지-대응-모니터링 단계간의 지속적 연동을 통한 이상거래 탐지 및 신규 패턴 획득 지원 기능
	기존 시스템과의 연동	- 기존 거래방식(사용자인증→거래지시→거래확정)과 FDS 기본 기능간의 상호 연계 가능

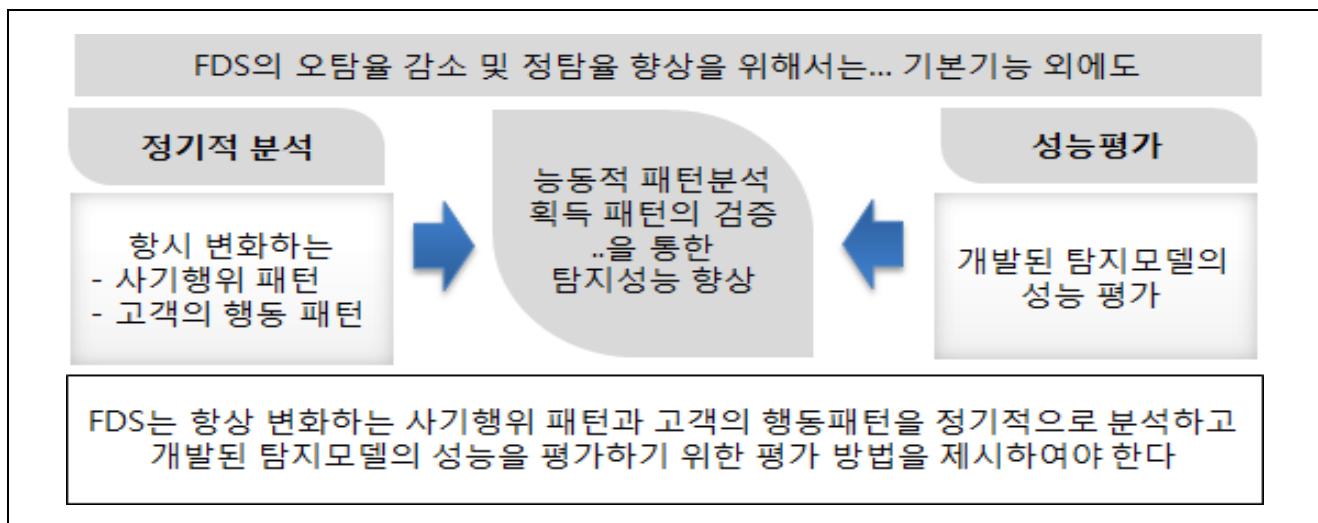
5. FDS 고려사항

가. 도입 및 운영 측면 고려사항

도입 측면 고려사항	운영 측면 고려사항	보안 측면 고려사항
<ul style="list-style-type: none">- 적용대상 산정 (서비스 유형/중요도/사용율 고려)- 구축방안 고려 (운영환경, 데이터종류, 개발환경)	<ul style="list-style-type: none">- 운영방식- 인력 배치/교육- 정기 분석 일정 (FDS기능향상과 HW가용성)	<ul style="list-style-type: none">- 데이터 보호 방안 수립 (기밀성/무결성 위한 ACL수립)- 감사 증적 데이터 유지

- FDS시스템 도입을 위한 대상산정 및 구축방안에 대한 검토와 도입 후 운영을 위한 운영방식 및 인력배치, 그리고 수집/분석데이터의 보호방안 수립에 대한 검토가 요구됨.

나. FDS성능 측면 고려사항



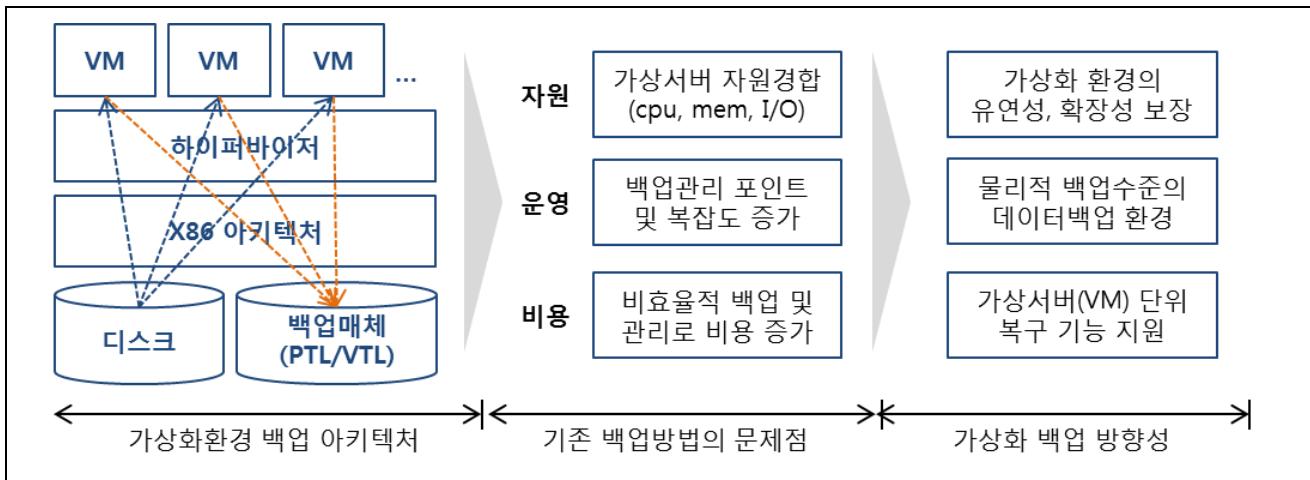
- FDS의 오탐율(오거부율, 오인식율)의 0%화를 위해 항시 변화하는 사기패턴 및 고객의 거래패턴을 지속적-능동적으로 분석하여 신규 패턴을 생성하고 이러한 패턴에 대한 성능평가 방법을 정의하고 지속적으로 수행하여야 함.

종 목	문 제
컴퓨터시스템응용기술사	4교시 6번

가상화 환경에서 데이터 백업방법 2가지를 설명하시오

도메인/토픽	컴퓨터구조 > 가상화 컴퓨터구조 > 보조기억장치 > 백업
키워드	Guest기반 백업, Proxy기반 백업, 증복제거(De-duplication), HSM, ILM, VTL/PTL, LTO
풀이 가이드	<ol style="list-style-type: none">가상화 환경에서의 기존 데이터 백업방법의 문제점가상화 환경에서의 백업방법 2가지가상화 환경에서 백업환경 구축 전략 <p>[문제에서 물어본 2번에 대해서 다양한 관점으로 풍부하게 작성해야 합니다.]</p>
필수항목 /연관토픽	가상화, Hypervisor, 백업 아키텍처, De-duplication 프라이빗 클라우드, DRaaS
참고문현	[특집]움트는 가상화 백업복구 시장, BIKorea, 2012. 7 (http://www.bikorea.net/news/articleView.html?idxno=6190) VMware환경에서의 백업 및 복구 솔루션, 기술자료
Advice	가상화 환경 및 Private Cloud 구축 관련 실무 문제입니다. 가상화 아키텍처의 이해를 바탕으로 기존 백업방법에 대한 문제점을 유추할 수 있어야 합니다. 수많은 가상서버 내에서의 백업수행은 자원경합을 유발하고, 백업데이터의 증가와 낭비 등의 문제를 야기합니다. 당분간, Private Cloud 심화토픽 및 실무형 문제가 출제되리라 예상됩니다.
작성자	108회 컴퓨터시스템응용기술사 / 임종범 (jblimn@gmail.com)

1. 가상화 환경에서의 기존 데이터 백업방법의 문제점



- 가상화 환경은 유연성과 확장성을 보장하는 동시에, 물리자원을 공유하는 아키텍처 구조적 특성상 기존의 백업/복구 체계는 자원경합, 운영의 비효율, 비용증가 등의 문제점을 초래함
- 가상화의 이점을 보장하는 동시에 백업복구의 단순화가 필요하며, 이는 물리 수준의 백업체계를 지원하는 동시에 논리(VM)수준의 복구체계를 지원하는 기술아키텍처가 요구됨

2. 가상화 환경에서의 백업방법 2가지

가. 백업실행 위치에 따른 백업방법 2가지

구분	Guest(VM)기반 백업	Proxy서버 기반 백업
개념	- 전통적인 물리적 서버 백업방식으로 VM 서버 백업 에이전트를 통한 백업 방식	- 백업대상 서버 이미지를 프록시 서버에 마운트하여 백업을 대행하는 백업 방식
구성도		
구성요소	<ul style="list-style-type: none"> - 백업 에이전트 - 백업 매체(PTL, VTL) 	<ul style="list-style-type: none"> - 스냅샷 이미지 - 프록시 서버(백업 대행 서버) - 백업매체 (PTL, VTL)
장점	- 백업구성 복잡도가 낮음	- 백업 워크로드 분산, 성능 저하 최소화

	<ul style="list-style-type: none"> - 백업환경 구축비용 낮음 	<ul style="list-style-type: none"> - 온라인 이미지 백업 및 중복제거 - 중앙집중백업으로 관리편의성 증가
단점	<ul style="list-style-type: none"> - 백업으로 인한 I/O 성능 하락 - 백업 관리포인트 증가 	<ul style="list-style-type: none"> - 백업 구성 복잡도 상승 - 백업 구축비용 상승
활용	<ul style="list-style-type: none"> - 중소규모 시스템 환경 - Web, App 서버와 같이, 백업사이즈가 작은 서버군 	<ul style="list-style-type: none"> - 중대형 시스템 환경 - DB서버와 같이 백업 사이즈가 크고, 성능에 민감한 서버군

- 게스트서버에서 직접 백업방법과 가상화 이미지를 마운트하여 백업을 대행하는 프록시 방식

나. 백업수준 및 복구대상에 따른 백업방법 2가지

구분	이미지 백업	데이터(파일) 백업
개념	<ul style="list-style-type: none"> - 하이퍼바이저에서 관리되는 가상서버(VM)의 이미지 파일을 백업하는 방식 	<ul style="list-style-type: none"> - 전통적인 물리백업과 동일하게, 가상서버(VM)내 파일단위로 백업하는 방식
개념도		
백업대상	<ul style="list-style-type: none"> - 하이퍼바이저가 인식하는 VM서버의 전체 이미지 파일 - VM서버 전체 이미지로 가상화솔루션에 종속적임 	<ul style="list-style-type: none"> - VM서버 내의 파일시스템 기반의 백업대상 파일 - 전통적인 물리백업 대상과 동일
장점	<ul style="list-style-type: none"> - 특정 시점으로 복구 가능 - 전체 시스템 단위의 백업으로 관리포인트 감소 	<ul style="list-style-type: none"> - 파일 단위의 복구 가능 - 백업 목적, 용도별 백업주기 다양화
단점	<ul style="list-style-type: none"> - 백업 사이즈 증가 - 파일단위의 개별 복구 불가 	<ul style="list-style-type: none"> - 백업 관리포인트 증가 - 백업파일을 통한 시스템환경 구성 시, 복잡도 증가
활용	<ul style="list-style-type: none"> - Web, App, DB서버 초기 이미지 템플릿 - 시스템 패치(Patch), 소스반영 등의 시스템 변경 전 백업 	<ul style="list-style-type: none"> - 파일 수준의 용도별 백업 및 복구 - Human Error 등으로 서버 파일 중 일부만 복구 필요 시 활용

	- 스냅샷을 통한 시점 백업	
--	-----------------	--

- 복구대상이 가상화서버(VM) 또는 데이터(파일)에 따라 백업 대상 및 방법이 달라짐

3. 구축유형에 따른 백업방법 2가지

구분	중앙집중식 백업	분산 백업
개념	- 백업환경을 백업 에이전트 중심의 가상화 서버 또는 프록시서버 레벨에서 SW 와 HW를 일체형으로 통합구축하는 방식	- 데이터 센터 수준의 분산된 백업환경으로 원격지의 백업센터로 데이터를 전송하는 방식
구성도		
구성요소	<ul style="list-style-type: none"> - 미디어 서버 - 백업 에이전트 - 백업매체(VTL, PTL) 	<ul style="list-style-type: none"> - 원격지 데이터 전송 프로토콜(FCIP, iSCSI) - 통신구간 암호화 - 중앙집중식 백업환경
활용	<ul style="list-style-type: none"> - 센터 단위의 백업환경 구축 - 전통적 백업 및 DR데이터 복제 환경 	<ul style="list-style-type: none"> - Virtual 데이터 센터 수준의 가상화 환경에서의 백업환경 구축 - DR as a Service 제공 및 사용

- 데이터 센터수준의 가상화환경의 확대로, 분산된 백업환경 및 서비스(DRaaS)가 대두됨

4. 가상화 환경에서 백업환경 구축 전략

구분	구축 전략	설명
고성능	Proxy 서버	<ul style="list-style-type: none"> - Mission Critical한 OLTP 시스템의 성능을 보장하기 위해, 백업 수행 서버를 프록시 서버로 분리하여 I/O 워크로드를 분산
	백업 스케줄링	<ul style="list-style-type: none"> - 백업 수행은 온라인 거래 Peak-Time 및 백업대상 서버간 병목 현상 발생되지 않도록 백업 스케줄링 계획 및 검토
효율성	중복 제거 (De-duplication)	<ul style="list-style-type: none"> - 중복제거 백업을 수행하여 공간 효율적으로 백업매체를 사용하고, VM이미지 및 파일 수준의 중복제거 기능 사전 검증
효율성 (계속)	이미지 템플릿	<ul style="list-style-type: none"> - 서버 OS와 시스템군(Web, WAS, DB)에 따른 초기 이미지를 템플릿으로 관리하여 백업파일을 통한 신속한 시스템 구성 가능

	HSM (Hierarchical Storage Mgmt.)	- ILM(Information LifeCycle Managemet)관점에서 데이터의 효용성을 바탕으로 계층적으로 백업매체(디스크, VTL, PTL)를 선정
가용성	DRaaS (DR as a Service)	- 가상화 환경에서의 백업환경을 클라우드 컴퓨팅으로 확장하여 비즈니스연속성보장체계(BCM)를 서비스로 제공받는 구조 전략
	가상화 클러스터링 (Virtual Data Center)	- 데이터 센터 수준의 가상화를 통해 서비스를 Active-Active로 구성하여 고가용성 및 확장성을 보장하는 백업 최소화 전략

- 전통적인 물리수준의 백업환경의 장점은 보장하면서, 가상화 환경의 인프라의 유연성과 확장성을 보장하는 가상서버(VM) 단위의 복구가 가능한 인프라스트럭처 구축전략이 필요