



제128회 컴퓨터시스템응용기술사 해설집

2022.07.02



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

국가기술자격 기술사 시험문제

기술사 제 128 회

제 3 교시 (시험시간: 100 분)

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

※ 다음 문제 중 4 문제를 선택하여 설명하시오. (각 10 점)

1. 계층구조의 통신 프로토콜 설계 시 고려해야 할 다음 각 요소에 대하여 설명하시오.

- 가. 오류 제어
- 나. 흐름 제어
- 다. 데이터 전달 방식

2. TCP(Transmission Control Protocol)에 대하여 다음을 설명하시오.

- 가. TCP 헤더(header)
- 나. TCP 헤더에 포함된 제어 플래그(Control Flag)의 종류

3. 디지털 포렌식(forensic) 과정에서 실제로 증거를 수집하려면 네트워크, 시스템, 데이터베이스 및 응용 프로그램 등 다양한 분야에 관한 지식이 필요하다. 다음 각 분야에서 디지털 포렌식의 증거를 수집하기 위한 방법을 설명하시오.

- 가. 네트워크 증거 수집
- 나. 시스템 증거 수집
- 다. 응용 프로그램 증거 수집

4. 클라우드 서비스 제공자는 AIaaS(AI as a Service)를 통해 인공지능을 쉽게 이용할 수 있도록 다양한 API(Application Program Interface)를 제공한다. 다음 항목을 설명하시오.

- 가. AIaaS 개념 및 특징
- 나. AIaaS 구조
- 다. 아마존 웹서비스(AWS)와 렉스(Lex)

5. KS X ISO/IEC/IEEE 29119을 활용하여 다음을 설명하시오.

- 가. 테스트 설계 및 구현 프로세스
- 나. 테스트 설계 기법
- 다. ISO/IEC 25010에서 정의한 테스트 품질 특성과 명세기반 테스트 설계 기법간의 매팅

6. 관계 데이터 모델이 가지고 있는 개체 무결성(Entity Integrity)과 참조 무결성(Referential Integrity)에 대하여 설명하시오.

01	통신 프로토콜 설계		
문제	계층구조의 통신 프로토콜 설계 시 고려해야 할 다음 각 요소에 대하여 설명하시오. 가. 오류 제어 나. 흐름 제어 다. 데이터 전달 방식		
도메인	네트워크	난이도	하 (상/중/하)
키워드	FEC, BEC, Stop and Wait, Sliding Window, Simplex, Full Duplex, Half Duplex		
참고문헌	ITPE 기술사회 자료 참고 [데이터통신및네트워크]계층 구조의 개념 : 네이버 블로그 (naver.com)		
풀이기술사	NS반 백기현 기술사(제 122회 정보관리기술사 / onlyride@naver.com)		

I. 네트워크의 독립적인 모듈 설계, 계층구조의 통신 프로토콜 개요

계층구조 개념	개념도	특징
- 네트워크에서 독립적인 고유 기능을 수행하는 모듈들이 상하위의 계층 구조로 연결되어 동작하는 구조	<div style="text-align: center;"> <p>상위 계층</p> <p>하위 계층</p> </div>	<ul style="list-style-type: none"> - 간단하고 작은 시스템으로 재구성 - 설계하고 구현이 간편 - 표준 인터페이스 제공 - 구조 단순화 가능 - 구조 변경 시 계층의 모듈 교체로 영향 최소화 가능

- 프로토콜 설계 시 ① 오류제어, ② 흐름제어, ③ 데이터 전달방식의 고려가 필요

II. 전송데이터 오류 수정, 오류제어 설명

가. 오류제어의 정의 및 유형

개념	- 부가정보를 활용하여 데이터 전송 중 발생되는 오류를 검출하여 재전송을 요구(BEC)하거나, 수신 측에서 직접 수정(FEC)하는 메커니즘	
개념도		
오류 유형	데이터 변형 오류	- 데이터가 수신측에 깨져서 도착하는 변형 오류
	데이터 분실 오류	- 데이터가 수신측에 도착하지 못하는 분실 오류

오류제어 기법유형	FEC	- 수신 측에서 오류가 발견 시 해당 오류를 검출할 뿐만 아니라, 오류를 정정할 수 있도록 부가정보를 활용한 에러기법
	BEC	- 오류가 발생하면 수신측은 송신 측에 오류가 발생한 사실을 알리고 송신 측에 오류가 발생한 프레임을 재전송 요구방식(ARQ : Automatic Repeat request 방식)

나. 오류제어 세부기법

구분	오류제어	설명
FEC	Hamming 코드	<ul style="list-style-type: none"> - 코드의 전송 시 Parity bit을 이용해 발생하는 오류를 검출(Detection)할 수 있을 뿐만 아니라 오류 코드의 정정(Correction)이 가능한 코드 - 잉여 비트 개수 계산식 : $2^p \geq m + p + 1$ - Parity Code의 위치가 1, 2, 4, 8 ... 2^{n-1}이며 수신 단말이 오류 수정
	Reed-Solomon 코드	<ul style="list-style-type: none"> - 2진 순환부호로, 통신 경로상 발생하는 랜덤 오류 및 Burst Error 모두 수정 - 고속 전송(수백 Mbps) 가능, 심볼단위 부호화
	LDPC	<ul style="list-style-type: none"> - 부호 길이를 증가시켜 오류 정정 능력을 향상시킨 계산 복잡도 유지 패리티체크 행렬 기반 오류 정정 부호(5G Data채널 코딩 기술로 채택)
	BCH 코드	<ul style="list-style-type: none"> - 순환 부호에 일종으로 선형 블록 부호이며, 부호화, 복호화에 용이 - Burst Error에 대한 오류 정정 능력이 매우 좋음
	Convolution 코드	<ul style="list-style-type: none"> - 비블록 코드로 메모리(기억성)을 갖는 부호화 - CEMA 무선환경에서 사용
	Turbo 코드	<ul style="list-style-type: none"> - Convolution 코드들 중에서 쉽게 부호화 할 수 있는 것들을 조합하여 랜덤하게 부호를 병렬 연접한 코드 - 반복 복호 알고리즘, 샤논의 이론적 채널용량 한계 근접 성능
BEC	Parity Check	<ul style="list-style-type: none"> - 짹수/홀수 패러티 비트를 데이터의 끝에 추가하여 전송 - 다중 Parity Error 검출이 어려움
	Check Sum	<ul style="list-style-type: none"> - 송신 데이터 전체를 합하고 이에 대한 1의 보수 값을 추가로 전송 - 수신 단말은 전체 수신 데이터를 합해서 1111..111 이 되는지 검증
	Block Sum	<ul style="list-style-type: none"> - 데이터를 블록으로 하여 Parity Check를 가로열과 세로열로 검출함 - Parity Check 방식에 비하여 잉여 비트의 용량이 상대적으로 커짐
	CRC	<ul style="list-style-type: none"> - N 개의 0 Bit를 삽입하고 이진 나눗셈을 수행하여 Remainder를 추가하여 송신하며 이진 나눗셈을 통해 나머지 계산으로 오류 검출
	Stop & Wait ARQ	<ul style="list-style-type: none"> - 송신 단말이 하나의 프레임을 전송하고 수신 단말은 이에 대한 오류 검출을 수행하여 그 결과를 ACK, NAK로 회신한 후 다음 프레임 전송
	Go back N ARQ	<ul style="list-style-type: none"> - 송신 단말은 Window Size만큼의 N개 프레임을 전송하고, 수신 단말은 N번째 오류 프레임을 회신하고 해당 N번째 이후 전제 재전송
	Selective Repeat	<ul style="list-style-type: none"> - Go back N ARQ가 N번째 프레임 이후의 전체를 재전송하는 반면에 Selective Repeat ARQ는 해당 N번째 프레임만 재전송 수행함.
	Adaptive ARQ	<ul style="list-style-type: none"> - 프레임의 오류 빈도의 통계적 결과에 따라 프레임의 길이를 조절

- 데이터링크 계층은 위에 언급한 제어기법 이외에도 하나의 회선/채널을 사용하는 환경에서 안정적인 송수신을 제공하기 위하여 접근제어 기법(CSMA/CD, CSMA/CA)을 사용

III. 용량이상 데이터 전송 방지 서비스, 흐름제어 설명

가. 흐름제어의 정의

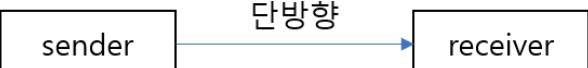
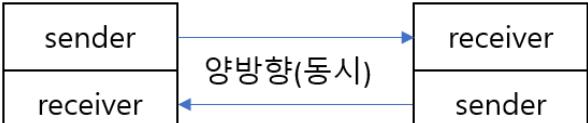
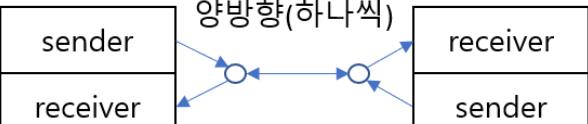
- 송신측 전송속도가 수신측 데이터 처리속도 보다 빠를 때 데이터 전송량 조절하는 속도 조절 기법
- Stop and Wait나 Sliding Window을 이용하여 송신자와 수신자 간의 속도차 보상

나. 흐름제어 방법

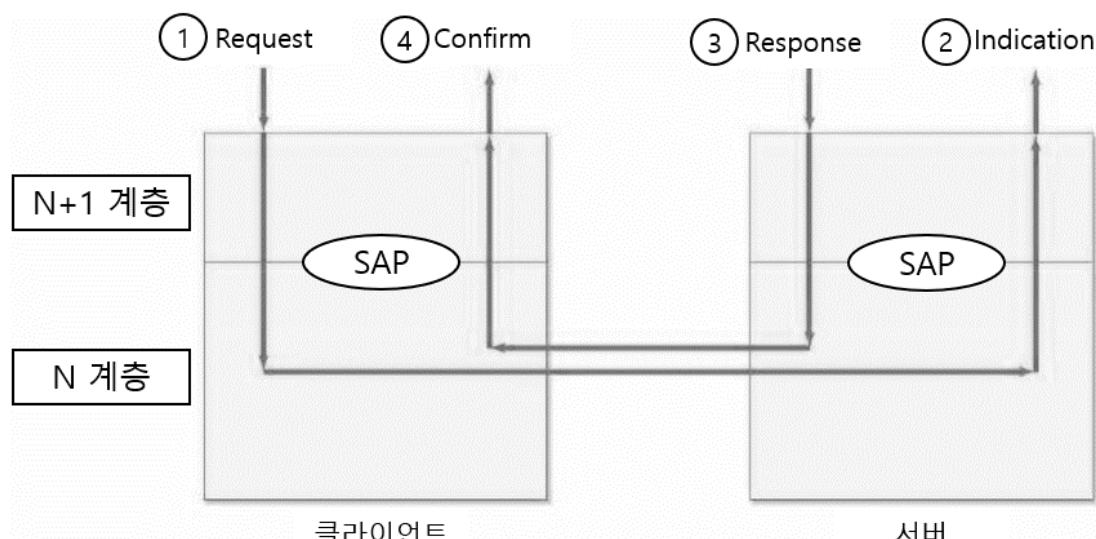
구분	정의 및 장단점	동작원리
Stop-and-Wait	<ul style="list-style-type: none"> - 송신 측이 프레임을 전송한 후 각 데이터 프레임에 대한 ACK를 기다려서, 이에 대한 ACK 프레임이 도착하면 그 후에 다음 프레임을 전송하는 기법 - 구조가 간단하여 구현 용이 - 송신측 내 최대 프레임 크기의 하나의 버퍼만 필요 - 전송 효율이 떨어짐 - 확인 응답을 기다려야 하므로 프레임 전송간의 오버헤드가 커짐 	<p>ACK, NAK, Wait Time</p>
Sliding Window	<ul style="list-style-type: none"> - 흐름제어를 위한 ARQ(Automatic Repeat Request)의 일종으로 일정한 윈도우 크기 이내에서 한번에 여러 패킷을 송신하고 이를 패킷에 대해 단지 한 번의 ACK 신호로써 수신 확인 - 전송한 프레임에 대한 ACK 프레임을 수신하지 않더라도 여러 개의 프레임 연속 전송 - 전송측과 수신측 양쪽에서 프레임을 보관하고 관리 - 오류발생이 가능하므로 Go-Back-N, 선택적 재전송 등 해결 방안 필요 	<p>RWND, CWND, Open, Close</p>

- 수신 단말이 허용 가능한 속도보다 송신 단말이 더 빠르게 프레임을 전송하지 않도록 흐름의 정도를 제어하는 기능

IV. 네트워크 설계 시 데이터 전달방식

유형	개념도	설명
단방향 (Simplex)		<ul style="list-style-type: none"> - 송신 측만 데이터 전달이 가능한 방식 - 단방향 (입출력장치)
반이중 (Half Duplex)		<ul style="list-style-type: none"> - 송신측은 데이터를 하나씩 전달 가능한 방식 - 쌍방향 (무전기 방식)
전이중 (Full Duplex)		<ul style="list-style-type: none"> - 데이터를 동시에 전달 가능한 방식 - 양방향 (이더넷 통신)

V. 계층구조에서 서비스 전달 방식, 서비스 프리미티브(Service Primitive)

구분	설명			
개념	<ul style="list-style-type: none"> - N계층을 기준으로, N계층인 서비스 제공자와, 상위계층인 N+1 계층의 서비스 이용자간의 상호 동작 			
개념도				
구성요소	서비스 제공자	- N+1 계층으로 서비스를 넘겨주는 N 계층		
	서비스 이용자	- N 계층으로부터 서비스를 받는 N+1 계층		
	SAP	- Service Access Point, 인접층간의 인터페이스		
신호종류 (방향)	요구(Request)	- 클라이언트가 서버에 서비스 요구		
	지시(Indication)	- 서버에 서비스 요구가 도착했음을 통지		
	응답(Response)	- 서버가 클라이언트에 서비스 응답		
	확인(Confirmation)	- 클라이언트에 응답이 도착했음을 통지		

주소표현	서비스 제공 계층	- L: Link Layer, N: Network Layer, T: Transport Layer, S: Session Layer
	수행되는 동작 이름	- CONNECT, DATA 등
	프리미티브 방향	- Request, Indication, Response, Confirmation
	파라미터	- 주소, 사용자 데이터, 원하는 서비스 형태, 데이터 최대크기 등

- 서비스 제공계층과 동작 이름등을 포함하여 서비스 프리미티브 표현 가능

“”

02	TCP(Transmission Control Protocol)		
문제	TCP(Transmission Control Protocol)에 대하여 다음을 설명하시오. 가. TCP 헤더(header) 나. TCP 헤더에 포함된 제어 플래그(Control Flag)의 종류		
도메인	네트워크	난이도	상 (상/중/하)
키워드	URG, ACK, PSH, RST, SYN, FIN		
참고문헌	ITPE 기술사회 서브노트		
풀이기술사	강남 평일 야간반 기술사(제 114회 정보관리기술사 / nikki6@hanmail.net)		

I. 연결 지향(Connection-oriented) 프로토콜, TCP 의 개요

가. 전송 제어 프로토콜, TCP(Transmission Control Protocol)의 정의

- 두 개의 호스트를 연결하고 데이터 스트림을 교환하게 해주는 중요한 네트워크 프로토콜

나. TCP 의 주요 특징

주요 특징	세부 내용
스트림 지향성	<ul style="list-style-type: none"> - 사용자 데이터를 바이트 기본 단위로 하는 데이터 스트림 형태로 처리 - 전송 측에서의 데이터 순서가 최종 목적지에서도 일치하도록 서비스 제공
가상 회선 연결	<ul style="list-style-type: none"> - 프로토콜 소프트웨어 모듈은 네트워크를 통해 메시지를 전송 준비 - 세부사항들에 대한 준비가 완료되면, 논리적인 연결 설정 과정을 수행
전 이중 전송 연결	<ul style="list-style-type: none"> - 두 개의 독립적인 스트림의 흐름으로 구성, 양방향 데이터 전송 수행

- 정확하게 데이터가 도착했음을 확인하기 위해 사용되는 하나의 프로시저 형태

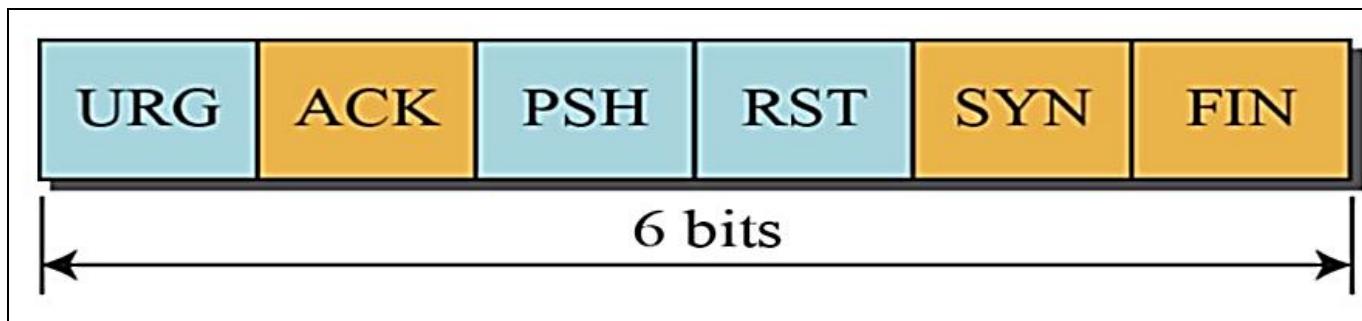
II. 출발지와 목적지의 정보를 지닌, TCP 헤더(Header) 상세 설명

구분	설명
정의	<ul style="list-style-type: none"> - 출발지, 목적지 Port 정보 및 제어 플래그 필드를 포함하는 20 Byte 크기를 가진 헤더 값
TCP 세그먼트 내 헤더 구성	<p>The TCP header fields are:</p> <ul style="list-style-type: none"> Source Port (16 bits) Destination Port (16 bits) Sequence number (32 bits) Acknowledgement number (32 bits) HLEN (4 bits) Options (6 bits) Control flags (URG, ACK, PSH, RST, SYN, FIN) Window size (16 bits) Checksum (16 bits) Urgent pointer (16 bits) Options and Padding <p>Note: 20 byte 헤더 (무옵션시) 옵션 포함하면 최대 60 bytes</p>

- TCP 헤더는 6 개의 Control Flag 필드들이 있음
- 이들은 논리적인 TCP 연결회선 제어 및 데이터 관리를 위해 사용

III. TCP 헤더에 포함된 제어 플래그(Control Flag) 종류

가. 제어 플래그(Control Flag) 구성도



나. 제어 플래그 종류 별 상세 설명

제어 플래그	의미	세부 내용
URG(Urgent)	- Urgent Pointer 필드에 값이 채워져 있음을 알림	- 송신측 상위 계층이 긴급 데이터라고 알려주면, 긴급 비트 URG를 1로 설정하고 순서에 상관없이 먼저 송신
ACK(Acknowledgement)	- 확인 응답 필드에 확인 응답번호 (Acknowledgement Number) 값이 셋팅 됐음을 알림	- 1로 셋팅 되면, 확인번호 유효함을 뜻함 - 0로 셋팅 되면, 확인번호 미포함 (즉, 32 비트 크기의 확인응답번호 필드 무시)
PSH(Push)	- 버퍼링된 데이터를 가능한한 빨리 상위 계층 응용프로그램에 즉시 전달할 것	- 수신 측은 버퍼가 찰 때까지 기다리지 않고, 수신 즉시 버퍼링된 데이터를 응용프로그램에 전달
RST(Reset)	- 연결확립(ESTABLISHED) 된 회선에 강제 리셋 요청	- 강제 리셋: RST=1 - 연결 상의 문제를 발견한 장비가 RST 플래그를 '1'로 설정한 TCP 세그먼트를 송출
SYN(Synchronize)	- TCP 연결설정 초기화를 위한 순서번호의 동기화	- 연결 요청: SYN=1, ACK=0 (SYN 세그먼트) - 연결 허락: SYN=1, ACK=1 (SYN+ACK 세그먼트) - 연결 설정: ACK=1 (ACK 세그먼트)
FIN (Finish)	- 송신기가 데이터 보내기를 끝마침	- 종결 요청: FIN=1 (FIN 세그먼트) - 종결 응답: FIN=1, ACK=1 (FIN+ACK 세그먼트)

- 3개 비트 플래그(RST,SYN,FIN)는 TCP 연결 설정 및 TCP 연결 종료에 주도적으로 사용

IV. 제어 플래그를 이용한 TCP 연결 설정 및 종료 사례

연결 성립(3 Way-Handshake)	연결 종료(4 Way-Handshake)
<pre> graph LR HostA[Host A] -- "Send SYN message" --> HostB[Host B] HostB -- "Receive SYN" --> HostA HostA -- "Send (SYN-ACK) message" --> HostB HostB -- "Receive ACK" --> HostA </pre>	<pre> graph LR HostA[Host A] -- "Send (FIN) message" --> HostB HostB -- "Receive FIN" --> HostA HostA -- "Send (ACK) message" --> HostB HostB -- "Send (FIN) message" --> HostA HostA -- "Receive ACK" --> HostB </pre>
<ol style="list-style-type: none"> 클라이언트는 서버에 접속을 요청하는 SYN(a) 패킷을 보냄 서버는 클라이언트의 요청인 SYN(a)을 받고 클라이언트에게 요청을 수락한다는 ACK(a+1)과 SYN(b)가 설정된 패킷을 클라이언트에게 발송 클라이언트는 서버의 수락 응답인 ACK(a+1)과 SYN(b) 패킷을 받고 ACK(b+1)을 서버로 보내면 연결이 성립(establish) 	<ol style="list-style-type: none"> 클라이언트가 연결 종료하겠다는 FIN 플래그 전송 서버는 클라이언트의 요청(FIN)을 받고 알겠다는 확인 메시지로 ACK를 보냄 데이터를 모두 보내고 통신이 끝났으면 연결이 종료되었다고 클라이언트에게 FIN 플래그를 전송 클라이언트는 FIN 메시지를 확인했다는 메시지(ACK)를 서버에게 보냄 클라이언트 ACK 메시지 받은 서버는 연결 종료

- 연결 종료 단계의 마지막 Closing 후 TIME-WAIT 단계가 존재
- TIME-WAIT: 새 연결과 겹치지 않도록 일정 시간 동안 기다린 후 최종 Closed 전이

“끝”

03	디지털 포렌식(forensic) 증거 수집		
문제	<p>디지털 포렌식(forensic) 과정에서 실제로 증거를 수집하려면 네트워크, 시스템, 데이터베이스 및 응용 프로그램 등 다양한 분야에 관한 지식이 필요하다. 다음 각 분야에서 디지털 포렌식의 증거를 수집하기 위한 방법을 설명하시오.</p> <p>가. 네트워크 증거 수집 나. 시스템 증거 수집 다. 응용 프로그램 증거 수집</p>		
도메인	디지털서비스	난이도	상 (상/중/하)
키워드	<p>디지털 포렌식 수집 절차 네트워크 증거 수집 – 스위치(switch), 허브(hub), libcap, BPF(Berkeley Packet Filter) 시스템 증거 수집 – 시스템 기본 정보, 시간 정보, 사용자 정보, Windows/Linux 명령어 응용 프로그램 증거 수집 – 윈도우 작업 관리자, top</p>		
참고문헌	<p>디지털 증거 수집 기술(서울과학기술대학교 박종혁 교수 강의 자료. 2015.2학기) [프로젝트]네트워크 포렌식 증거 수집(https://net711.tistory.com/entry/프로젝트-네트워크-포렌식-증거-수집) 디지털 증거분석 표준 가이드라인에 대한 연구(백석대학교 이성진 교수 저) ITPE 124회 대비 Final Round 3일차 해설집</p>		
풀이기술사	안경환 기술사(제 110회 정보관리기술사 / akh.itpe@gmail.com)		

I. 디지털 기기의 범죄 흔적 Tracking. 디지털 포렌식의 개요

정의	<p>- 컴퓨터를 이용하거나 활용해 이뤄지는 범죄 행위에 대한 법적 증거 자료 확보를 위해 컴퓨터 시스템, 네트워크 등 가상 공간의 자료가 법적 증거물로 법원에 제출될 수 있도록 확보하는 일련의 절차와 방법</p>				
	사전 준비	증거 수집	증거의 포장 및 이송	조사 분석	보고서 작성
디지털 포렌식 원칙	<ul style="list-style-type: none"> ● 디지털 기기 및 데이터의 유형 숙지 ● 디지털 포렌식 교육 및 연구 개발 	<ul style="list-style-type: none"> ● 수집 대상 파악 ● 압수 대상 선정 ● 증거 목록 작성 ● 물리적 증거 수집 ● 관련자 면담 ● 문서화 	<ul style="list-style-type: none"> ● 압수물 개별 포장 ● 전자파 및 증거 방지 포장 ● 증거를 포장 및 운반 	<ul style="list-style-type: none"> ● 데이터 이미징 ● 데이터 추출 및 분류 ● 데이터 조사 및 증거 검색 ● 정밀 검토 	<ul style="list-style-type: none"> ● 용서 설명 ● 객관적 설명 ● 결과 정리

	- 재현(再現)의 원칙	- 동일조건에서는 동일결과 도출, 법정현출(顯出)
	- 신속성의 원칙	- 모든 과정은 지체 없이 신속하게 진행
	- 연계 보관성(Chain of custody)	- 증거물 획득, 이송, 분석, 보관, 법정제출의 각 단계에서 담당자 및 책임자가 명확
	- 무결성(Integrity)의 원칙	- 디지털 데이터는 위·변조와 훼손이 용이하기 때문에 수집 이후 변경되지 않았음을 입증
디지털 포렌식 증거 수집 대상	<ul style="list-style-type: none"> - 네트워크, 시스템, 응용 프로그램 - 디지털 기기의 다양성으로 인해 각 유형별 디지털 증거 수집의 방법이 다르게 진행 	

II. 네트워크 증거 수집 방법

가. 패시브(Passive) 증거 수집 방법

개념	- 계층 이상의 상위 계층에서 데이터를 송신(transmission)하지 않고 포렌식(forensic) 증거 수집을 하는 방법 (예: 트래픽 수집)	
기법	대상 및 기법	설명
물리적 감청	- cable, 탭(Tap)	<ul style="list-style-type: none"> - 동축 케이블, 괌성유 등을 대상으로 signal 수집 - 인라인 네트워크 탭, 뱼파이어 탭, 유도코일, 광섬유 탭
	- radio 주파수 - 802.11 지원하는 Network 어댑터	<ul style="list-style-type: none"> - Broadcast SSID, 무선 AP의 mac 주소, 연결된 Client의 MAC 주소 등을 수집
	- 허브(Hub)	<ul style="list-style-type: none"> - Network Physical Layer 허브(Hub) 장비로 수신되는 모든 프레임(frame)을 수신하여 수집
	- 스위치(Switch) - 스니핑(Sniffing)	<ul style="list-style-type: none"> - Network DataLink Layer 장비로 Traffic을 스니핑(snipping) - 포트 미러링(Port Mirroring), MAC 플러딩(MAC flooding), ARP 스푸핑(ARP spoofing)
트래픽 수집 소프트웨어	- libpcap, WinPcap	<ul style="list-style-type: none"> - 네트워크 인터페이스에 대해 데이터 링크 계층의 프레임을 캡처하고 필터링
	- BPF(Berkeley Packet Filter)	<ul style="list-style-type: none"> - 데이터링크에 대한 원시 인터페이스를 제공하여 원시 링크 계층 패킷을 송수신 - 필터링을 통해 필요한 데이터 수집
	- tcpdump	<ul style="list-style-type: none"> - 네트워크 트래픽을 Capture, Filtering, Analysis 하기 위한 도구
	- Wireshark	<ul style="list-style-type: none"> - 트래픽을 캡처하고, 필터링하고, 분석하기 위해 설계된 그래픽 기반의 오픈소스 툴

나. 액티브(Active) 증거 수집 방법

개념	- 네트워크 상의 스테이션(station)과 상호작용하여 증거를 수집하는 방법 - 예) 콘솔(console) 또는 네트워크 인터페이스를 통해 포트 스캔(Port Scan)을 수행	
기법	대상 및 기법	설명
일반 인터페이스 이용	- Console, Serial Port	- Network 장비의 Serial Port를 통해 터미널에서 콘솔(Console)로 연결

	- SCP, SFTP	- SCP(Secure Copy)와 SFTP(SSH File Transfer Protocol) 은 SSH프로토콜을 이용한 커맨드 라인 상에서 파일을 전송 - 원격 서버의 중요 파일(예: /etc/passwd)을 현재 Local PC로 복사
	- SNMP(Simple Network Management Protocol)	- IP 네트워크상의 장치로부터 정보를 수집 및 관리하는 Protocol - network 장비의 점검에 많이 이용되며 많은 tool들이 SNMP 기반으로 제작
직접 접근 없는 조사	- 포트 스캐닝(Port Scanning) - nmap	- nmap과 같은 도구를 사용 - 열려 있는 포트와 장비의 소프트웨어 버전 정보를 획득
	- 취약점 스캐닝 (Vulnerability Scanning)	- 포트 스캐닝에 더해 취약점을 테스트 진행 - 대상 시스템이 침해당했을 가능성이 의심될 경우, 어떤 경로를 통해 침해되었는지 파악하기 위한 단서 제공

III. 시스템 증거 수집 방법

가. 시스템 운용 여부에 따른 수집 방법

유형	분석 대상 및 수집 방법	설명
비운용 시스템의 증거 획득 - 비활성 시스템	- 디스크(Disk) 복제 · 디스크(Disk) 이미지 획득	- 대상 저장매체의 모든 내용을 섹터 단위로 읽어 분석 시스템에 파일로 저장
	- 이동식 저장 매체	- 쓰기 잠금 기능 지원 시, 쓰기 금지 설정 후 데이터 수집
운용 시스템의 증거 획득 - 활성 시스템	- 탐침(probe) 하드웨어 또는 소프트웨어	- 하드웨어(주로 USB 케이블)와 드라이버 소프트웨어를 설치한 후에 이를 통하여 필요한 자료를 획득(예: memory dump)
	- 별도의 분석 소프트웨어 사용	- 대상 시스템이 웜바이러스에 감염되었거나, 또는 해킹 당하여 시스템에 있는 실행파일들이 더 이상 정상적이지 못하다고 판단될 때 사용
	- 대상 시스템에 내장되어 있는 소프트웨어 사용	- ls, dd 등 시스템 점검을 위해 기본 내장된 프로그램을 이용

나. 시스템 수집 정보 유형에 따른 수집 방법

정보 유형	분석 대상 및 수집 방법	설명
기본 정보	- OS, Version, Resource - Install Package - 디스크 정보, 파티션 정보	- 운영체제 관련 Log 및 configuration 정보 확보 - /proc/version: 커널 버전 정보 - /proc/cpuinfo: CPU 정보 - /proc/meminfo: 메모리 사용량 - /proc/stat: 시스템 상태
시간 정보	- local Time, 파일 MAC 시간 - 시스템 구동 시간	- 증거 수집 시, 시간이 기준 - 사용 명령어: date, uptime

사용자 정보	<ul style="list-style-type: none"> - 계정 정보(전체 계정 목록, 현재 로그인한 계정) - Remote User Account, Remote IP, 실제 사용자 식별 정보 	<ul style="list-style-type: none"> - 수집 시스템의 현재 계정 정보 확보 - who: 현재 로그인 상태인 사용자 정보 확인 - w: 각 사용자가 진행 중인 작업 확인 - last: 최종 로그인한 사용자 정보
--------	---	---

IV. 응용 프로그램 증거 수집 방법

가. 범용 / 서버 응용 프로그램 증거 수집 방법

증거 수집 대상	증거 수집 도구	설명
프로세스 정보	<ul style="list-style-type: none"> - 윈도우 작업 관리자, Process Explorer - top 	<ul style="list-style-type: none"> - 실행 중인 프로세스 정보를 수집 및 분석하여 비정상 프로세스를 분석
	<ul style="list-style-type: none"> - 프로세스 구조체(eProcess) 	<ul style="list-style-type: none"> - 덤프(dump)한 물리 메모리에서 프로세스 구조체를 추출 - 은닉 프로세스 탐지, 이전 실행 프로세스 목록 추출
프로그램 설정 파일	<ul style="list-style-type: none"> - Register Editor - log file 	<ul style="list-style-type: none"> - 응용 프로그램 환경을 구성하는 정보가 저장된 Register와 구동 결과가 나오는 log 파일 정보

나. 임베디드 디바이스(Embedded Device) 응용 프로그램 증거 수집 방법

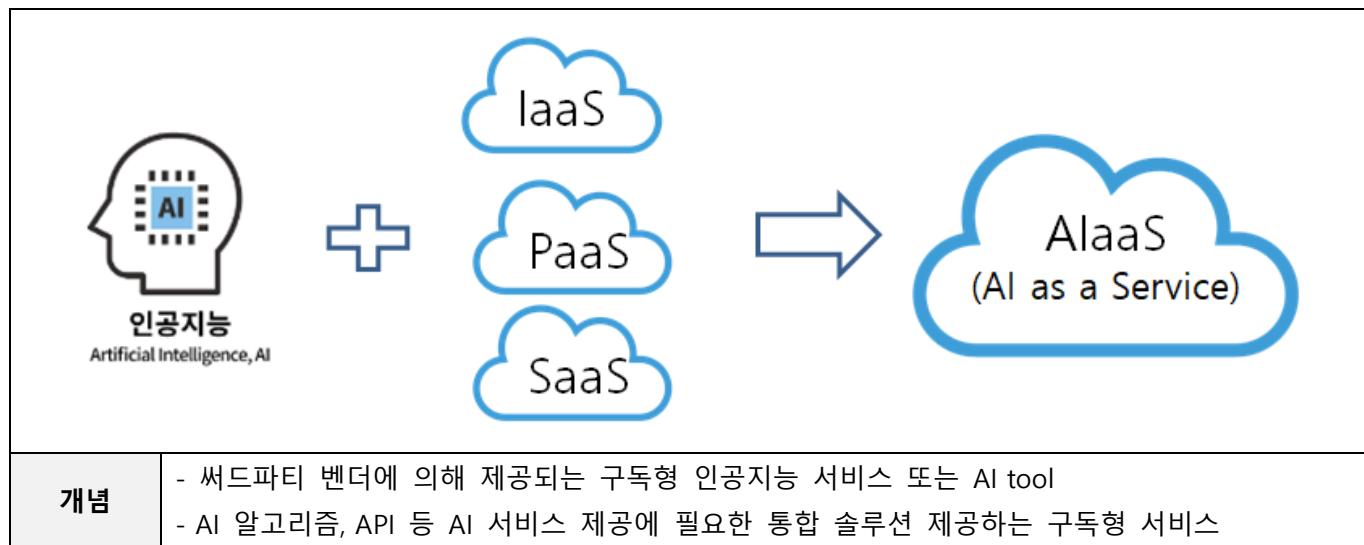
증거 수집 기법	증거 수집 도구	설명
파일 단위 복사	- USB 통신 프로토콜	- USB 통신 프로토콜 또는 제조사에서 제공하는 파일 접근 USB 명령을 통해 파일을 복사
제조사 전용 통신 프로그램	- PC Sync Software	- 제조사에서 제공하는 별도 컨텐츠(content) 접근 프로그램을 이용하여 파일 단위 수집 기법
설정 백업	- 백업 소프트웨어	- PC와 연동하여 설정과 데이터를 백업 수행

“끝”

04	AlaaS(AI as a Service)		
문제	클라우드 서비스 제공자는 AlaaS(AI as a Service)를 통해 인공지능을 쉽게 이용할 수 있도록 다양한 API(Application Program Interface)를 제공한다. 다음 항목을 설명하시오. 가. AlaaS 개념 및 특징 나. AlaaS 구조 다. 아마존 웹서비스(AWS)와 렉스(Lex)		
도메인	디지털서비스	난이도	상 (상/중/하)
키워드	AI, 클라우드, 서비스, 플랫폼, IaaS, PaaS, SaaS, AI Data, AI Compute, Serverless Computing, AI Service, Cognitive Computing, Conversational AI, Framework, CSP, Lex		
참고문헌	ITPE 기술사회 서브노트 AlaaS (AI as a Service) 서비스와 활용 고려사항 서비스형 인공지능(AlaaS)의 개념 및 도입 효과 (appier.com) https://docs.aws.amazon.com/ko_kr/lexv2/latest/dg/what-is.html		
풀이기술사	NS반 차상인 기술사(제 125회 컴퓨터시스템응용기술사 / itpe.ince@gmail.com)		

I. 서비스형 인공지능, AlaaS(AI as a Service)의 개념 및 특징

가. AlaaS(AI as a Service)의 개념



- 기능별 서비스 유형별 특징이 존재하며, 최근 금융, 공공, 의료 산업을 중심으로 빠르게 도입사례 증가 추세

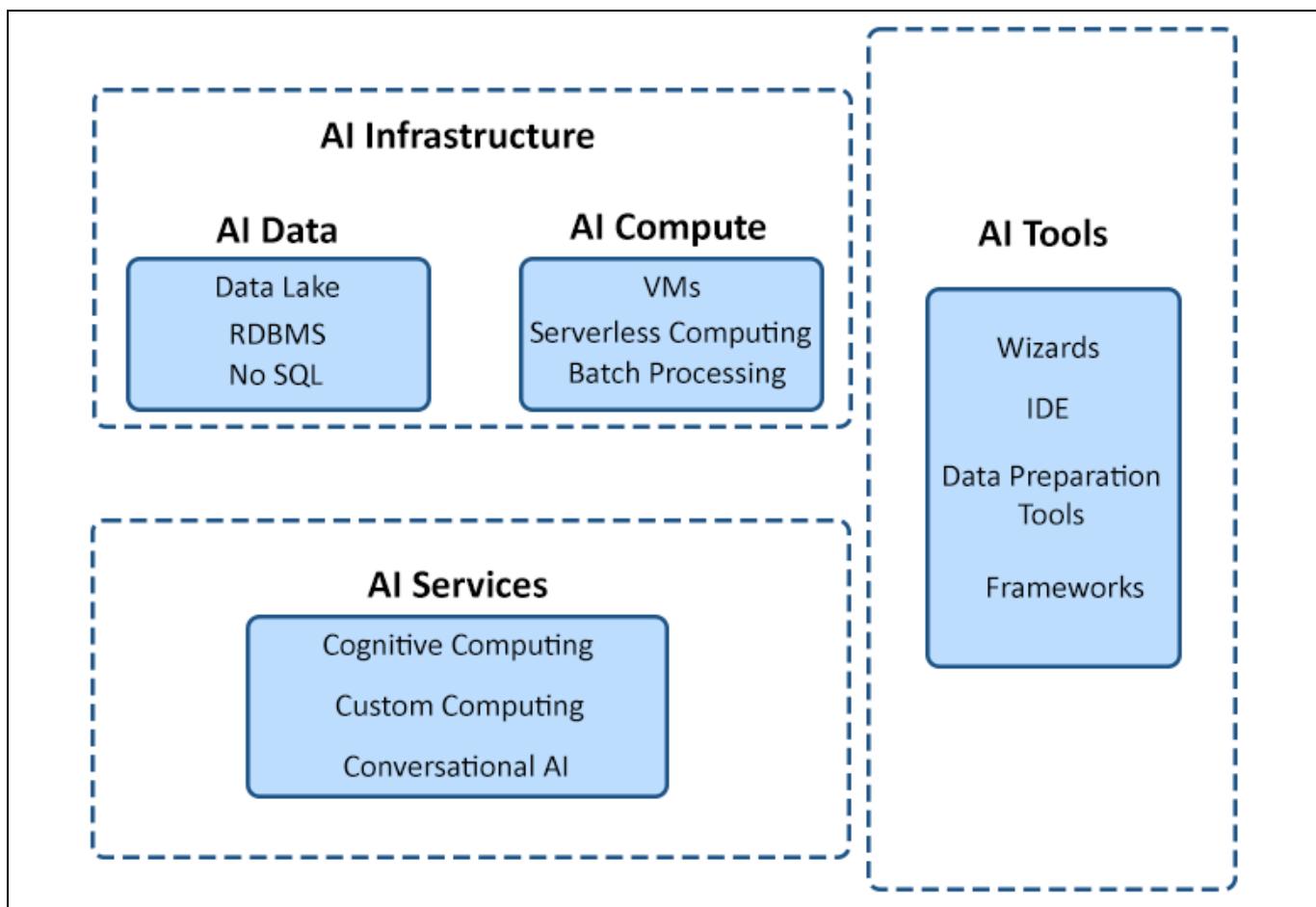
나. AlaaS의 특징

구분	특징	설명
서비스 측면	- H/W 인프라 서비스	<ul style="list-style-type: none"> - 인공지능 시스템 구축을 위한 하드웨어 인프라를 서비스 형태로 제공 - GPU, NPU 등의 하드웨어 인프라 On-Demand 형태로 제공
	- 인지 컴퓨팅 API 서비스 제공	<ul style="list-style-type: none"> - API 서비스를 이용해 NLP, 컴퓨터 비전, 지식 매핑, 지능형 검색, 번역 및 감정 감지 기능 처리 가능.
	- 머신러닝(ML) 프레임워크 서비스 제	<ul style="list-style-type: none"> - 방대한 양의 데이터가 없는 소규모 기업에서 AI 모델 구축 가능 - 빅데이터 분석, 패턴 분석 등 인공지능 엔진 프레임워크 제공

	공	
비즈니스 측면	- 완전 관리형 머신 러닝 서비스	- 개발자가 자체 AI 모델을 구축할 필요 없이, AlaaS 솔루션의 사전 구축된 모델, 사용자 지정 템플릿 및 코드 없는 인터페이스를 이용 가능
	- 높은 비용 효율	- 경비 지출 및 관련 위험을 피하면서 필요한 AI 기능을 활용 가능 - 도메인 업무에 집중하면서 AI 기능 서비스 형태로 연계 가능
	- 투자 위험 감소	- 낮은 CAPEX로 투자 실패 시의 위험 부담 감소 가능
	- 신규 시장 진출	- 전문 인력 및 인프라 구축 없이 인공지능 기술 활용 가능 - AI 서비스 지원을 통해 다양한 서비스 분야 시장 진출 가능

II. AlaaS의 구조

가. AlaaS 구조도



나. AlaaS 구조 구성요소

구분	구성요소	설명
AI Infrastructure	AI Data	- Data Lake, RDBMS, NoSQL 등에 수집된 모든 유형의 데이터들이 ML 모델에 입력 값으로 활용
	AI Compute	- VM, Serverless Computing 및 Batch Processing 기술이 포함 - 병렬 처리 향상 및 ML 작업을 자동화 하는데 사용

AI Services	Cognitive Computing	- API를 통해 음성, 텍스트 분석, 음성 번역 및 검색 기능을 지원 - 단일 REST API 호출로 애플리케이션이 통합
	Custom Computing	- 사용자 지정 데이터 셋을 사용하여 인지 컴퓨팅 구축 가능토록 지원 - 데이터에 따른 알고리즘 선택 및 ML 모델 훈련을 지원
	Conversational AI	- 챗봇 및 가상비서 서비스를 제공 및 플랫폼 간 봇 통합 지원
AI Tools	Wizards	- 전문 인력 없이 ML 모델 훈련 설정을 돋는 마법사 기능 제공 - Multi-tenant 형태로 개발 환경에서 작동
	IDE	- ML 모델 테스트 및 관리를 위한 IDE 지원
	Data Preparation Tools	- ETL 작업을 수행할 수 있는 데이터 전처리 도구 지원 - 전처리 된 데이터 ML 훈련 데이터 셋으로 파이프라인에 공급
	Frameworks	- TensorFlow, Apache MXNet 및 Torch 등의 프레임워크와 함께 On-Demand 형태의 VM Template 제공

- 대표적 CSP 사업자 중 AWS(Amazon Web Service)에서도 AlaaS 서비스 제공

III. 아마존 웹서비스(AWS)와 렉스(Lex)

가. 아마존 웹 서비스(AWS) 설명

구분	특징	설명
개념	- 컴퓨팅 파워, 데이터 베이스, 스토리지, 콘텐츠 전송 등의 다양한 클라우드 컴퓨팅 서비스를 제공하는 CSP	
지원 서비스	- Compute	- EC2 (Elastic Compute Cloud) : 서버 인스턴스 - Lightsail : EC2보다 간소화된 VPS 서비스 - Lambda(Serverless Computing) : Serverless Computing
	- Networking & Content Delivery	- CloudFront : CDN 서비스 - Route 53 : DNS 서비스 - AWS App Mesh : Microservice Service Mesh - API Gateway : HTTP Proxy, 인증, 로드밸런싱, 라우팅
	- Storage	- S3 : Simple Storage Service. - S3 Glacier : 백업 전용 스토리지. - Elastic File System : 블록 스토리지 NFS기반.
	- Database	- RDS : MySQL과 유사한 RDBMS 서비스 - Aurora : RDS에 Auto Scaling 기능을 추가한 형태 - DynamoDB : NoSQL 데이터베이스 - ElastiCache : 인 메모리 데이터베이스 - Neptune : 그래프 데이터베이스
	- Management & Governance	- CloudWatch : 로그(Log) 관리 및 모니터링 서비스 - CloudFormation : 클라우드 구성 배포 서비스 - CloudTrail : 사용자 활동 및 API 추적 서비스
	- Machine Learning	- Lex : 텍스트 & 음성인식 기반 대화형 인공지능 서비스

		- Rekognition : 지능형 이미지 분석 서비스.
--	--	---------------------------------

- 300여개 이상의 클라우드 서비스 보유, 최근 AlaaS인 Lex를 서울 리전에 출시하며 한국어 음성 지원 시작

나. AWS의 대표적 AlaaS, 렉스(Lex) 설명

구분	설명								
개념	<ul style="list-style-type: none"> - 대화형 인터페이스를 애플리케이션에 설계, 구축, 테스트, 배포하기 위해 고급 자연어 모델을 사용하는 완전관리형 인공 지능 서비스 								
개념도	<pre> graph LR A[Amazon Lex
Build conversational interfaces into any app using voice & text] --> B[Define your chat experience
What issue/problem are you solving for and what are the desired outcomes] B --> C[Intent: Desired action] B --> D[Utterance: User input] B --> E[Prompt: Request data] B --> F[Slot: Required data] B --> G[Fulfillment: Completed action] C --> H[Build and deploy your chatbot
Test your chatbot before deploying into production] </pre> <p>The diagram illustrates the Amazon Lex architecture. It starts with a box labeled "Amazon Lex Build conversational interfaces into any app using voice & text". An arrow points to a central box labeled "Define your chat experience" with the sub-instruction "What issue/problem are you solving for and what are the desired outcomes". From this central box, five arrows point to separate boxes: "Intent: Desired action", "Utterance: User input", "Prompt: Request data", "Slot: Required data", and "Fulfillment: Completed action". Finally, an arrow points from the "Intent" box to a box labeled "Build and deploy your chatbot Test your chatbot before deploying into production", which features an icon of a rocket launching.</p>								
사용사례	<table border="1"> <tr> <td>- 가상 상담원 및 음성 도우미 구축</td> <td>- 가상 고객 센터 상담원 및 대화형 음성 응답을 통해 셀프 서비스 기능을 지원</td> </tr> <tr> <td>- 정보 응답 자동화</td> <td>- FAQ 응답 제공하는 대화형 솔루션 설계</td> </tr> <tr> <td>- 애플리케이션 봇</td> <td>- 애플리케이션에서 기본적인 사용자 태스크를 자동화 지원 - Lambda 통해 다른 엔터프라이즈 SW에 원활한 연결 가능</td> </tr> <tr> <td>- 상담 기록 활용</td> <td>- 기존 고객 센터 상담 기록 이용 Chatbot 설계</td> </tr> </table>	- 가상 상담원 및 음성 도우미 구축	- 가상 고객 센터 상담원 및 대화형 음성 응답을 통해 셀프 서비스 기능을 지원	- 정보 응답 자동화	- FAQ 응답 제공하는 대화형 솔루션 설계	- 애플리케이션 봇	- 애플리케이션에서 기본적인 사용자 태스크를 자동화 지원 - Lambda 통해 다른 엔터프라이즈 SW에 원활한 연결 가능	- 상담 기록 활용	- 기존 고객 센터 상담 기록 이용 Chatbot 설계
- 가상 상담원 및 음성 도우미 구축	- 가상 고객 센터 상담원 및 대화형 음성 응답을 통해 셀프 서비스 기능을 지원								
- 정보 응답 자동화	- FAQ 응답 제공하는 대화형 솔루션 설계								
- 애플리케이션 봇	- 애플리케이션에서 기본적인 사용자 태스크를 자동화 지원 - Lambda 통해 다른 엔터프라이즈 SW에 원활한 연결 가능								
- 상담 기록 활용	- 기존 고객 센터 상담 기록 이용 Chatbot 설계								

- 조직의 인력과 기술력, 비즈니스 전략 등 고려하여 도입 유형 및 최상의 CSP 파트너 업체 선정 중요

“풀”

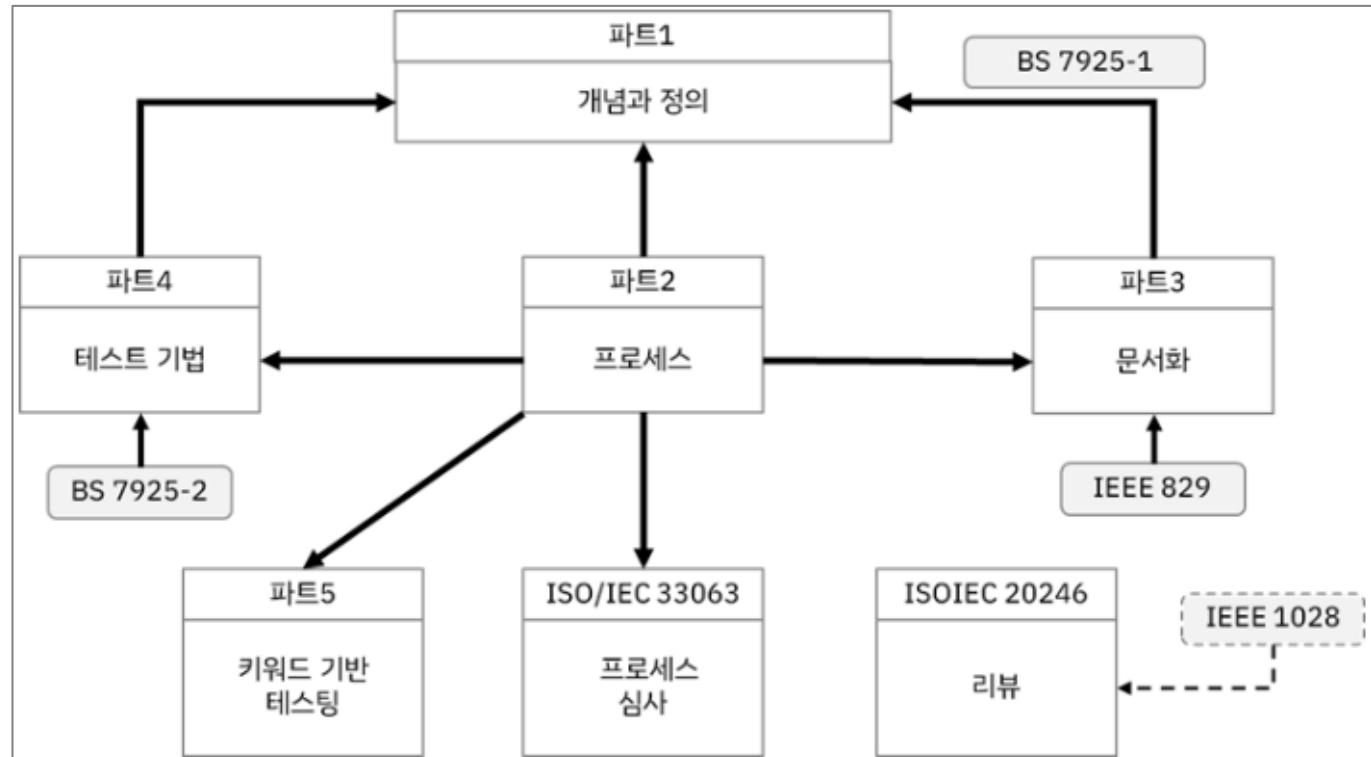
05	KS X ISO/IEC/IEEE 29119		
문제	KS X ISO/IEC/IEEE 29119을 활용하여 다음을 설명하시오. 가. 테스트 설계 및 구현 프로세스 나. 테스트 설계 기법 다. ISO/IEC 25010에서 정의한 테스트 품질 특성과 명세기반 테스트 설계 기법간의 매핑		
도메인	소프트웨어공학	난이도	중 (상/중/하)
키워드	파트1~5, 조직/테스트관리/동적테스트, 다계층 프로세스, 명세기반, 구조기반, 경험기반 동등분할, 분류트리기법, 경계값분석, 구문규칙테스팅, 조합테스트, 결정테이블테스팅, 구문,분기		
참고문헌	https://www.tta.or.kr/data/androReport/ttaJnal/167-5-3.pdf https://sun-kakao.tistory.com/263 소프트웨어 테스트 가이드, 정보통신진흥원(NIPA)		
풀이기술사	유술사PE (제 113회 컴퓨터시스템응용기술사 / itpe_you@naver.com)		

I. 소프트웨어 테스팅 표준, KS X ISO/IEC/IEEE 29119의 개요

가. KS X ISO/IEC/IEEE 29119의 정의

- '검증 및 확인(Verification and Validations)' 활동 중 '동적 테스팅(Dynamic Testing)'에 대한 절차와 기법을 다로는 국제 및 국내 국가 표준

나. KS X ISO/IEC/IEEE 29119의 구성



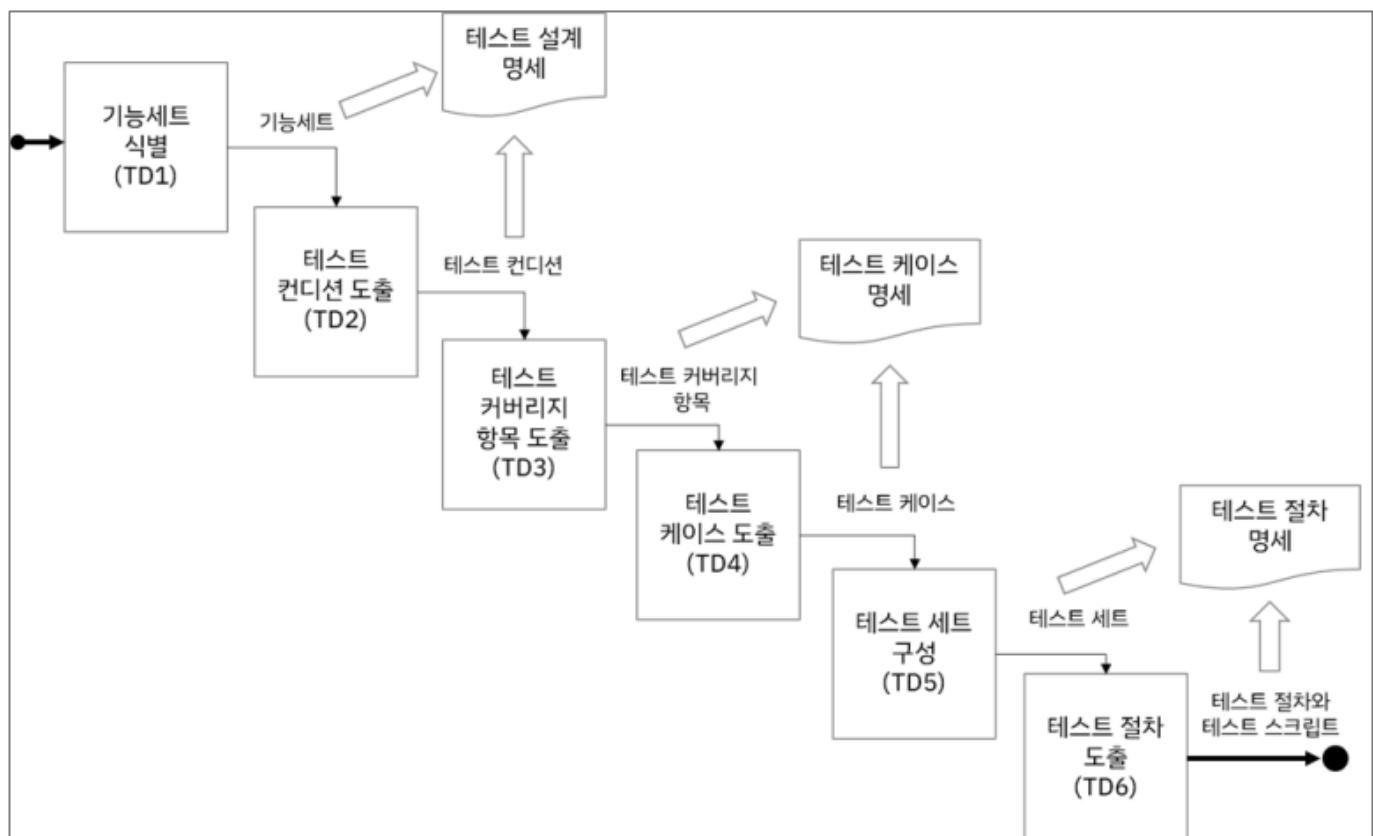
- Part1~5로 구성되며, V&V 활동의 범주 내에서 동적 테스팅과 관련된 절차와 기법을 제공하지만, 특정 생명 주기모델에만 한정된 제공하지 않음(폭포수 모델부터 애자일접근방법까지 두로 적용가능)

II. ISO 29119-2, 테스트 설계 및 구현 프로세스

가. 테스트 설계 및 구현 프로세스의 개념

- 테스트 프로세스에 관한 부분으로 조직/테스트관리/동적테스트의 세가지 수준의 다계층 프로세스 모델

나. 테스트 설계 및 구현 프로세스



III. ISO 29119-4, 테스트 설계 기법

가. 테스트 설계 기법의 개념

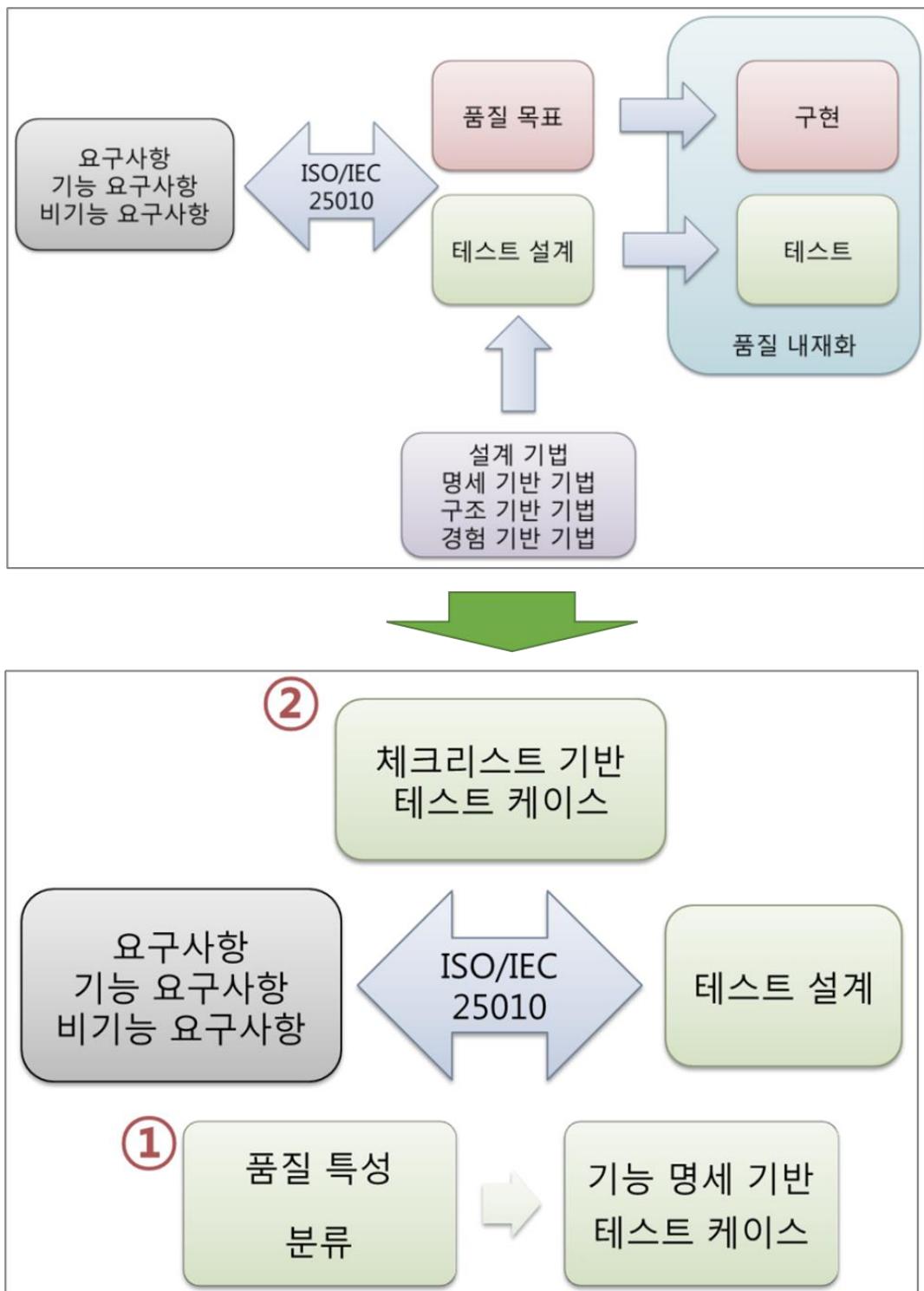
- 소프트웨어 테스팅 기법에 관한 부분으로 테스트 설계 및 구현 프로세에서 활용할 수 있는 명세기반 테스트 설계, 구조기반 테스트 설계, 경험 기반 테스트 설계 기법을 제공

나. 테스트 설계 기법의 유형



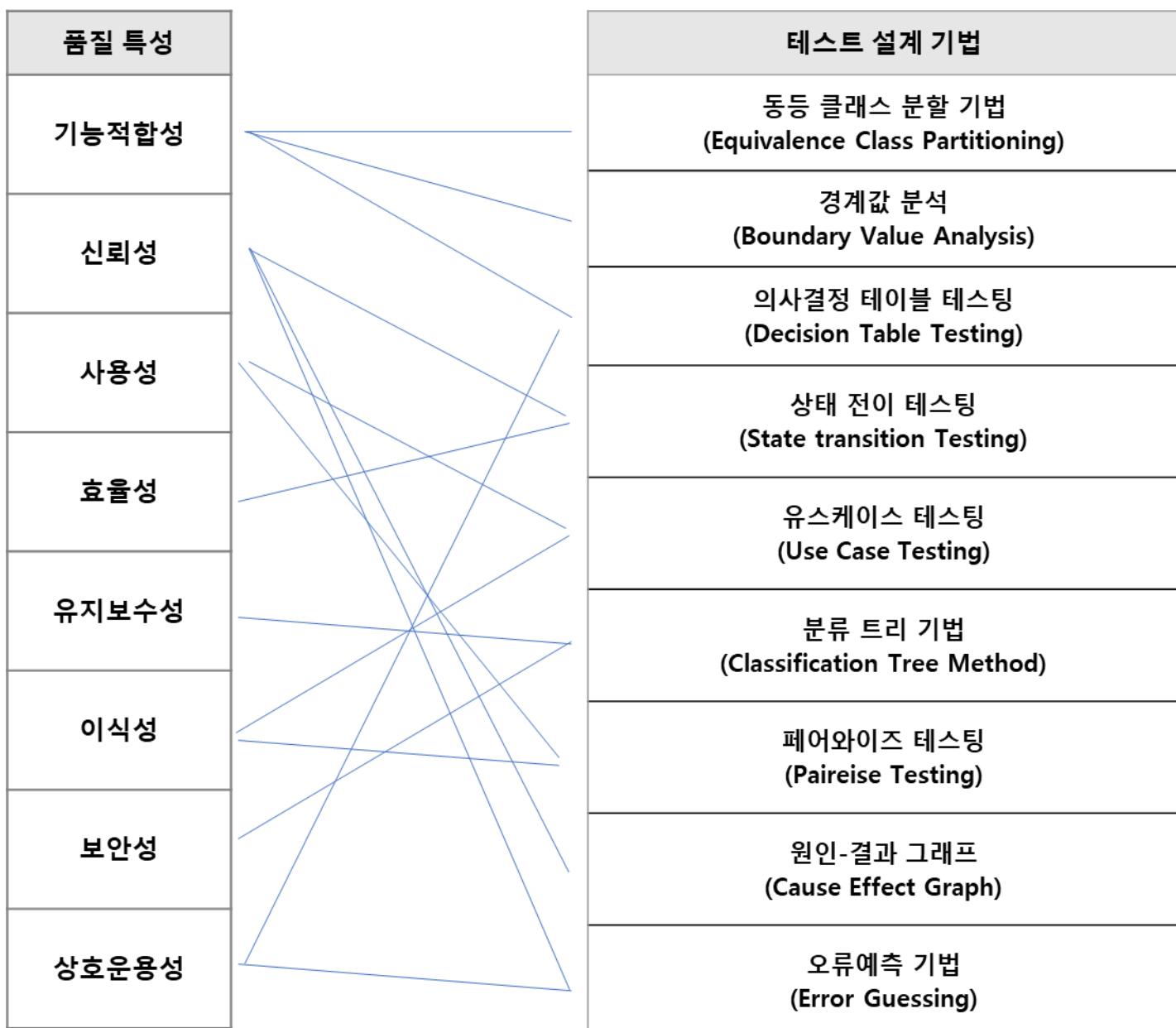
IV. ISO/IEC 25010에서 정의한 테스트 품질 특성과 명세기반 테스트 설계 기법간의 매핑

가. 테스트 품질 특성과 명세기반 테스트 설계 기법간의 매핑 관계



- 사용자의 소프트웨어에 제품 품질 요구사항을 반영하여 품질 목표를 정의하며, 품질 목표는 소프트웨어 제품의 검증 목표로 사용되며, ISO/IEC 25010을 기반으로 설계 기법을 도출하는 방법은 크게 두가지로 분류
- 첫번째로 기능 명세를 품질 특성 별로 분류하여 기능명세 기반 테스트 케이스를 도출하는 방법
- 두번째로 품질 특성 별 체크리스트를 기반으로 테스트 케이스를 도출하는 방법

나. 테스트 품질 특성과 명세기반 테스트 설계 기법간의 매핑



- 품질특성에 기능성/사용성 이외에도 효율성, 유지보수성, 신뢰성, 이식성, 보안성, 상호운용성 등 있으면 품질 목표와 요구사항을 통해 테스트 케이스와 명세기반 설계 기법을 적용

“끌”

[참고] 전자레인지 예시를 통한 테스트 품질 특성과 명세기반 테스트 설계 기법간의 맵핑

품질특성	품질목표 (전자레인지 예)	요구사항 (전자레인지 예)	명세기반 테스트 설계 기법	테스트 케이스
기능성	<ul style="list-style-type: none"> - 명세서에 명시된 기능이 정상적으로 동작하는 것을 보증한다 	<ul style="list-style-type: none"> - 사용자는 조리시간과 세기를 선택할 수 있다 - 키패드는 0~9까지의 숫자키가 있고, [Time], [Power] 버튼으로 값을 입력할 수 있다. 	<ul style="list-style-type: none"> - 페어와이즈테스팅 - 모든조합기법 	TC1 : 조리시간과 세기가 정확하게 설정되는지 확인한다
		<ul style="list-style-type: none"> - 키패드는 0~9까지의 숫자키가 있고, [Time], [Power] 버튼으로 값을 입력할 수 있다. - [Time] 선택 시 LED 표시창에 4 자리 숫자(분단위 2 자리, 초단위 2 자리. 0~99 분 사이)까지 입력할 수 있다. 	<ul style="list-style-type: none"> - 경계값분석 - 동등분할 	TC2 : [Time], [Power] 버튼으로 0~9까지 값을 입력할 수 있는지 확인한다.
사용성	<ul style="list-style-type: none"> - 사용자가 원하는 작업을 명확히 수행할 수 있도록 전자레인지은 사용자가 수행하는 작업을 명확하게 표시한다. 	<ul style="list-style-type: none"> - [Time] 선택 시 LED 표시창에 4 자리 숫자(분단위 2 자리, 초단위 2 자리. 0~99 분 사이)까지 입력할 수 있다. 	<ul style="list-style-type: none"> - 결정테이블테스팅 	TC3 : [Time] 선택 시 LED 표시창에 4 자리 숫자(분단위 2 자리, 초단위 2 자리. 0~99 분 사이)까지 입력할 수 있는지 확인한다.
			<ul style="list-style-type: none"> - 결정테이블테스팅 	TC4 : [Power] 선택 시 LED 표시창에 3 자리 숫자(0~800 사이)까지 입력 가능하며 입력값이 800을 넘어가면 그 값은 무시되고 800으로만 설정되는지 확인한다.
		<ul style="list-style-type: none"> - 상태전이테스팅 	<ul style="list-style-type: none"> - 상태전이테스팅 	TC5: [Power]를 입력하지 않을 경우의 기본값이 70으로 설정되는지 확인한다.

- 품질특성에 기능성/사용성 이외에도 효율성, 유지보수성, 신뢰성, 이식성, 보안성, 상호운용성 등 있으면 품질 목표와 요구사항을 통해 테스트 케이스와 명세기반 설계 기법을 적용이 가능

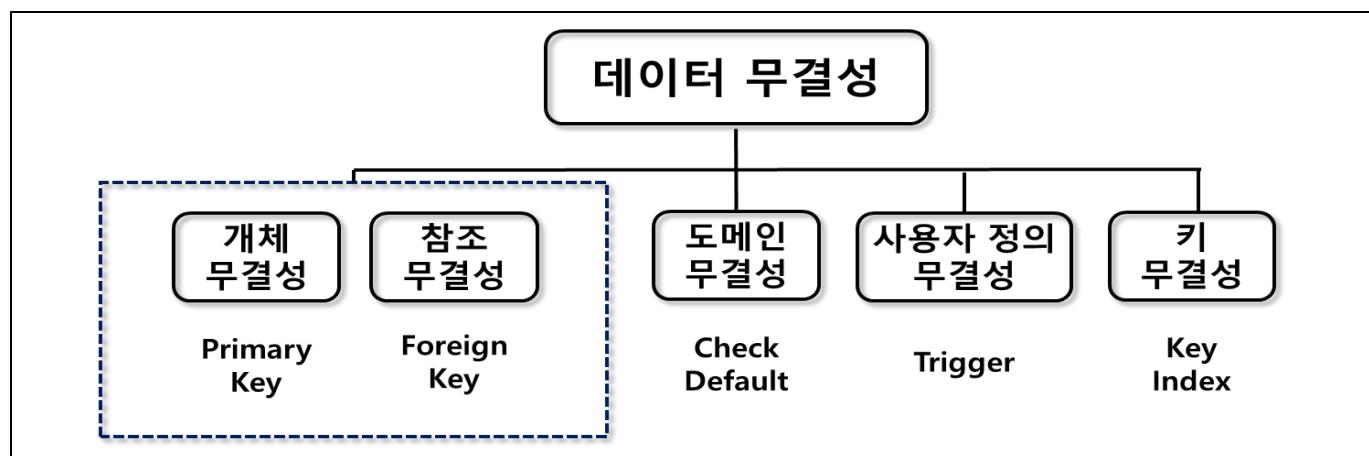
06	데이터 무결성		
문제	관계 데이터 모델이 가지고 있는 개체 무결성(Entity Integrity)과 참조 무결성(Referential Integrity)에 대하여 설명하시오.		
도메인	데이터베이스	난이도	하 (상/중/하)
키워드	Primary Key, Foreign Key, RESTRICTED, CASCADE, DEFAULT, NULL		
참고문헌	오라클로 배우는 데이터베이스 개론과 실습(박우창, 한빛미디어) 위키백과(https://ko.wikipedia.org/wiki/데이터_무결성)		
풀이기술사	김민 PE(제 120회 정보관리기술사 / itpe.min@gmail.com)		

I. 데이터 무결성의 개요

가. 데이터 무결성의 정의

- 데이터베이스의 중요한 기능으로 관계, 참조 무결성 제한이나 규칙에 의해 데이터의 정확성과 일관성을 유지하고 보증하는 성질

나. 데이터 무결성의 종류



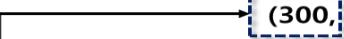
- 데이터 무결성의 대표적인 종류로 릴레이션 관계 표현과 관련된 개체 무결성과 참조 무결성 존재

II. 개체 무결성(Entity Integrity) 설명

가. 개체 무결성의 개념

개념	- 릴레이션은 고유성(Unique)이 만족되는 Primary Key를 가져야 하며 NULL 값을 허용하지 않는 특성															
개념도	<p style="text-align: center;">Primary Key</p> <table border="1"> <thead> <tr> <th>학번</th> <th>이름</th> <th>학과코드</th> </tr> </thead> <tbody> <tr> <td>100</td> <td>김민</td> <td>1521</td> </tr> <tr> <td>200</td> <td>박민</td> <td>1522</td> </tr> <tr> <td>300</td> <td>최민</td> <td>1521</td> </tr> <tr> <td>400</td> <td>김민</td> <td>1536</td> </tr> </tbody> </table>	학번	이름	학과코드	100	김민	1521	200	박민	1522	300	최민	1521	400	김민	1536
학번	이름	학과코드														
100	김민	1521														
200	박민	1522														
300	최민	1521														
400	김민	1536														

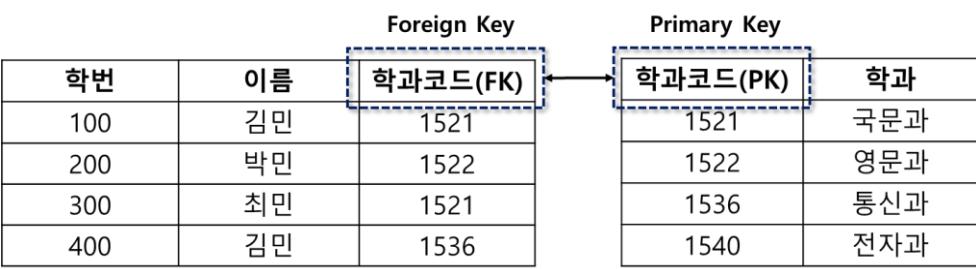
나. 개체 무결성의 제약조건

제약조건	예시	설명															
Primary Key 유일성	<p>유일성 위배</p>  <table border="1"> <tr><th>학번</th><th>이름</th><th>학과코드</th></tr> <tr><td>100</td><td>김민</td><td>1521</td></tr> <tr><td>200</td><td>박민</td><td>1522</td></tr> <tr><td>300</td><td>최민</td><td>1521</td></tr> <tr><td>400</td><td>김민</td><td>1536</td></tr> </table> <p>삽입 거부</p>	학번	이름	학과코드	100	김민	1521	200	박민	1522	300	최민	1521	400	김민	1536	<ul style="list-style-type: none"> - Primary Key 값이 기존 다른 값과 중복 되지 않는지 검사
학번	이름	학과코드															
100	김민	1521															
200	박민	1522															
300	최민	1521															
400	김민	1536															
NUL NULL 불가	<p>NULL 불가</p>  <table border="1"> <tr><th>학번</th><th>이름</th><th>학과코드</th></tr> <tr><td>100</td><td>김민</td><td>1521</td></tr> <tr><td>200</td><td>박민</td><td>1522</td></tr> <tr><td>300</td><td>최민</td><td>1521</td></tr> <tr><td>400</td><td>김민</td><td>1536</td></tr> </table> <p>삽입 거부</p>	학번	이름	학과코드	100	김민	1521	200	박민	1522	300	최민	1521	400	김민	1536	<ul style="list-style-type: none"> - Primary Key 값은 NULL 값 불가
학번	이름	학과코드															
100	김민	1521															
200	박민	1522															
300	최민	1521															
400	김민	1536															

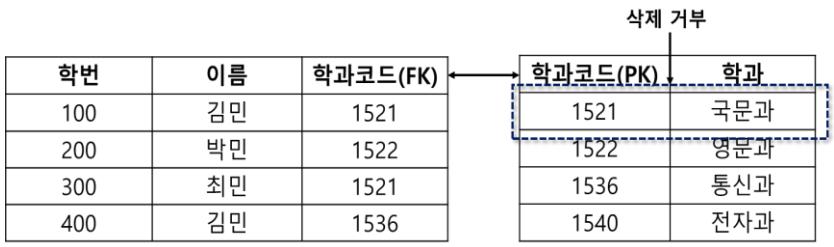
- 관계 데이터 모델이 투플을 삽입하거나 수정할 때 개체 무결성 제약조건을 확인

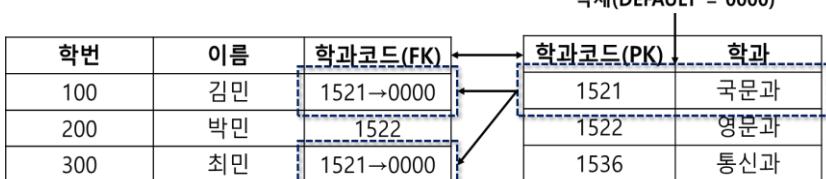
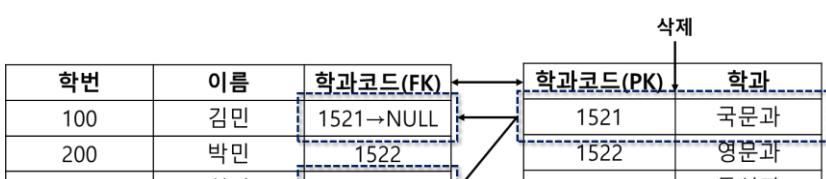
III. 참조 무결성(Referential Integrity) 설명

가. 참조 무결성의 개념

개념	<ul style="list-style-type: none"> - Foreign Key가 참조하는 다른 릴레이션의 Primary Key에 해당하는 값이 Primary Key 값이나 Null 허용 특성 																										
개념도		<p>Foreign Key</p> <table border="1"> <tr><th>학번</th><th>이름</th><th>학과코드(FK)</th></tr> <tr><td>100</td><td>김민</td><td>1521</td></tr> <tr><td>200</td><td>박민</td><td>1522</td></tr> <tr><td>300</td><td>최민</td><td>1521</td></tr> <tr><td>400</td><td>김민</td><td>1536</td></tr> </table> <p>Primary Key</p> <table border="1"> <tr><th>학과코드(PK)</th><th>학과</th></tr> <tr><td>1521</td><td>국문과</td></tr> <tr><td>1522</td><td>영문과</td></tr> <tr><td>1536</td><td>통신과</td></tr> <tr><td>1540</td><td>전자과</td></tr> </table>	학번	이름	학과코드(FK)	100	김민	1521	200	박민	1522	300	최민	1521	400	김민	1536	학과코드(PK)	학과	1521	국문과	1522	영문과	1536	통신과	1540	전자과
학번	이름	학과코드(FK)																									
100	김민	1521																									
200	박민	1522																									
300	최민	1521																									
400	김민	1536																									
학과코드(PK)	학과																										
1521	국문과																										
1522	영문과																										
1536	통신과																										
1540	전자과																										

나. 참조 무결성의 제약조건

제약조건	예시	설명																									
RESTRICTED	 <table border="1"> <tr><th>학번</th><th>이름</th><th>학과코드(FK)</th></tr> <tr><td>100</td><td>김민</td><td>1521</td></tr> <tr><td>200</td><td>박민</td><td>1522</td></tr> <tr><td>300</td><td>최민</td><td>1521</td></tr> <tr><td>400</td><td>김민</td><td>1536</td></tr> </table> <p>삭제 거부</p> <table border="1"> <tr><th>학과코드(PK)</th><th>학과</th></tr> <tr><td>1521</td><td>국문과</td></tr> <tr><td>1522</td><td>영문과</td></tr> <tr><td>1536</td><td>통신과</td></tr> <tr><td>1540</td><td>전자과</td></tr> </table>	학번	이름	학과코드(FK)	100	김민	1521	200	박민	1522	300	최민	1521	400	김민	1536	학과코드(PK)	학과	1521	국문과	1522	영문과	1536	통신과	1540	전자과	<ul style="list-style-type: none"> - 자식 릴레이션에서 참조하고 있을 경우 부모 릴레이션의 삭제 작업 거부
학번	이름	학과코드(FK)																									
100	김민	1521																									
200	박민	1522																									
300	최민	1521																									
400	김민	1536																									
학과코드(PK)	학과																										
1521	국문과																										
1522	영문과																										
1536	통신과																										
1540	전자과																										

CASCADE	 <p>학번 이름 학과코드(FK)</p> <table border="1"> <tr><td>100</td><td>김민</td><td>1521</td></tr> <tr><td>200</td><td>박민</td><td>1522</td></tr> <tr><td>300</td><td>최민</td><td>1521</td></tr> <tr><td>400</td><td>김민</td><td>1536</td></tr> </table> <p>학과코드(PK) 학과</p> <table border="1"> <tr><td>1521</td><td>국문과</td></tr> <tr><td>1522</td><td>영문과</td></tr> <tr><td>1536</td><td>통신과</td></tr> <tr><td>1540</td><td>전자과</td></tr> </table>	100	김민	1521	200	박민	1522	300	최민	1521	400	김민	1536	1521	국문과	1522	영문과	1536	통신과	1540	전자과	- 자식 릴레이션의 관련 투플을 같이 삭제 처리
100	김민	1521																				
200	박민	1522																				
300	최민	1521																				
400	김민	1536																				
1521	국문과																					
1522	영문과																					
1536	통신과																					
1540	전자과																					
 <p>학번 이름 학과코드(FK)</p> <table border="1"> <tr><td>100</td><td>김민</td><td>1521→0000</td></tr> <tr><td>200</td><td>박민</td><td>1522</td></tr> <tr><td>300</td><td>최민</td><td>1521→0000</td></tr> <tr><td>400</td><td>김민</td><td>1536</td></tr> </table> <p>학과코드(PK) 학과</p> <table border="1"> <tr><td>1521</td><td>국문과</td></tr> <tr><td>1522</td><td>영문과</td></tr> <tr><td>1536</td><td>통신과</td></tr> <tr><td>1540</td><td>전자과</td></tr> </table>	100	김민	1521→0000	200	박민	1522	300	최민	1521→0000	400	김민	1536	1521	국문과	1522	영문과	1536	통신과	1540	전자과	- 자식 릴레이션의 관련 투플을 미리 설정해둔 값으로 변경	
100	김민	1521→0000																				
200	박민	1522																				
300	최민	1521→0000																				
400	김민	1536																				
1521	국문과																					
1522	영문과																					
1536	통신과																					
1540	전자과																					
NULL	 <p>학번 이름 학과코드(FK)</p> <table border="1"> <tr><td>100</td><td>김민</td><td>1521→NULL</td></tr> <tr><td>200</td><td>박민</td><td>1522</td></tr> <tr><td>300</td><td>최민</td><td>1521→NULL</td></tr> <tr><td>400</td><td>김민</td><td>1536</td></tr> </table> <p>학과코드(PK) 학과</p> <table border="1"> <tr><td>1521</td><td>국문과</td></tr> <tr><td>1522</td><td>영문과</td></tr> <tr><td>1536</td><td>통신과</td></tr> <tr><td>1540</td><td>전자과</td></tr> </table>	100	김민	1521→NULL	200	박민	1522	300	최민	1521→NULL	400	김민	1536	1521	국문과	1522	영문과	1536	통신과	1540	전자과	- 자식 릴레이션의 관련 투플을 NULL 값으로 설정함
100	김민	1521→NULL																				
200	박민	1522																				
300	최민	1521→NULL																				
400	김민	1536																				
1521	국문과																					
1522	영문과																					
1536	통신과																					
1540	전자과																					

- 삭제 대상 투플이 NOT NULL일 경우 삭제 연산 거절

IV. 개체 무결성과 참조 무결성 비교

비교 항목	개체 무결성	참조 무결성
제약 대상	- 투플	- 속성과 투플
사용 키	- Primary Key	- Foreign Key
NULL 값	- 불가	- 허용
제약 조건	- 2개	- 4개
변경 영향 검토	- 투플 삽입	- 부모/자식 투플 삽입, 삭제

- 개체, 참조 무결성 외에 도메인 무결성 제약조건(Type, Check)도 고려하여 데이터 무결성 유지 필요

“끝”



ITPE

ICT 온라인, 오프라인 융합 No 1

PMP 자격증 정보관리기술사/컴퓨터시스템응용기술사
IT전문가과정 정보시스템감리사
정보통신기술사 애자일

오프라인 명품 강의

ITPE 기술사회

제128회 컴퓨터시스템응용기술사 기출문제 해설집

대상 정보관리기술사, 컴퓨터시스템응용기술사, 정보통신기술사, 정보시스템감리사 시험

발행일 2022년 07월 02일

집필 강정배PE, 안경환PE, 전일PE, 유술사PE, 김민PE, 백기현PE, 차상인PE

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.