



제132회 컴퓨터시스템응용기술사 해설집

2024.01.27



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

국가기술자격 기술사 시험문제

기술사 제 132 회

제 4 교시 (시험시간: 100 분)

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

※ 다음 문제 중 4 문제를 선택하여 설명하시오. (각 10 점)

1. 컴퓨터 시스템의 캐시(Cache) 메모리와 관련하여 아래 사항을 설명하시오.
 - 캐시메모리에서 주소 매핑(mapping)이 필요한 이유와 주소 매핑(mapping) 방식
 - 캐시 쓰기 정책(write policy)
 - 다중 프로세서 시스템에서 캐시 일관성(Cache Coherence)을 유지하기 위한 기법
2. 한국지능정보사회진흥원에서 제시한 클라우드 서비스 활용사업 감리 점검에 대하여 아래 사항을 설명하시오.
 - 공공부문의 클라우드 사업 유형
 - 클라우드 서비스 활용사업의 점검 단계 활동, 검토항목
3. 디지털 트윈 기술을 이용한 제조(정보통신단체표준, TTAK.KO-11.0284)에 대하여 아래 사항을 설명하시오.
 - 제조를 위한 디지털 트윈(digital twin) 기술 개념
 - 제조 디지털 트윈(digital twin) 프레임워크
 - 액터(actor) 정보 테이블의 항목 및 항목 값
4. 운영체제에서 발생할 수 있는 deadlock 현상에 대하여 아래 사항을 설명하시오.
 - deadlock 의 개념
 - deadlock 과 starvation 의 차이점
 - deadlock 이 발생하기 위한 조건 4 가지
 - deadlock 발생 시 처리 방안

5. 서포트 벡터 머신(Support Vector Machine)은 기계학습 분야에서 많이 활용되는 학습 모델이다. 아래 사항을 설명하시오.

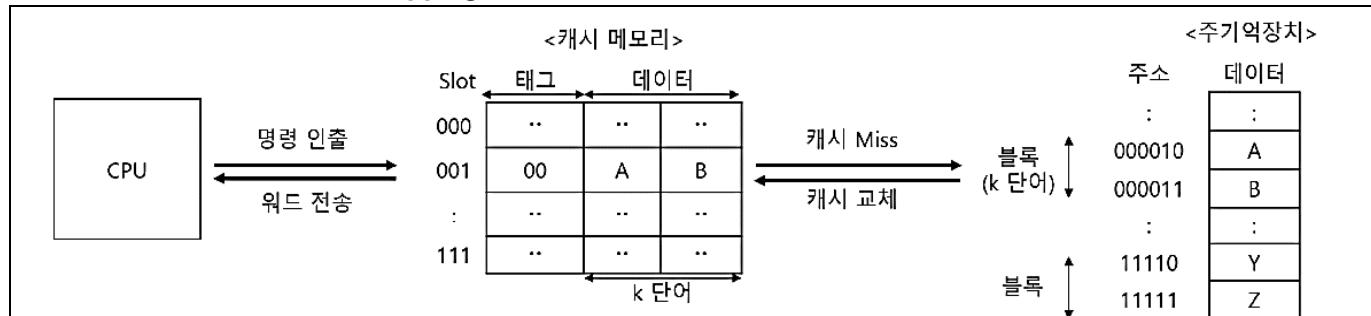
- 가. SVM의 개념
- 나. SVM의 동작방식
- 다. SVM의 장단점 및 활용사례

6. 초저지연 서비스와 부하분산을 위한 모바일 엣지 컴퓨팅(Mobile Edge Computing) 기술이 확대되고 있다. 아래 사항을 설명하시오.

- 가. 모바일 엣지 컴퓨팅의 개념
- 나. 모바일 엣지 컴퓨팅의 플랫폼 구조 및 구성요소
- 다. 모바일 5G에서 엣지 컴퓨팅 적용 방안

01	캐시(Cache) 메모리		
문제	<p>컴퓨터 시스템의 캐시(Cache) 메모리와 관련하여 아래 사항을 설명하시오.</p> <p>가. 캐시메모리에서 주소 매핑(mapping)이 필요한 이유와 주소 매핑(mapping) 방식</p> <p>나. 캐시 쓰기 정책(write policy)</p> <p>다. 다중 프로세서 시스템에서 캐시 일관성(Cache Coherence)을 유지하기 위한 기법</p>		
도메인	CA/OS	난이도	중(상/중/하)
키워드	완전-완관, 직접 사상, 세트-연관 사상		
출제배경	CA/OS 기본 토픽 숙지 확인		
참고문헌	ITPE 기술사회		
해설자	전일 기술사(제 114회 정보관리기술사 / roseemachine@naver.com)		

I. 캐시메모리에서 주소 매핑(mapping)이 필요한 이유

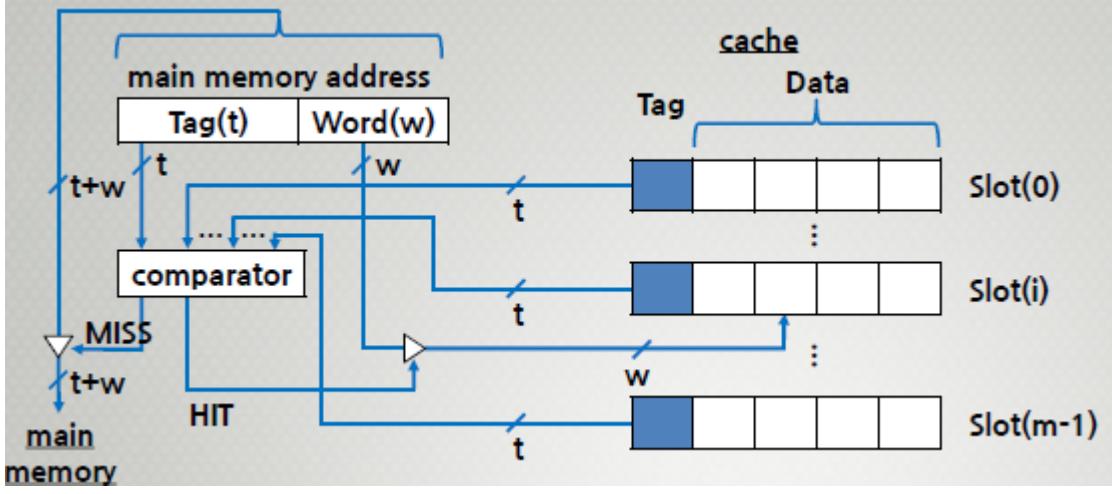


개념	- 캐시 메모리 Miss 발생시 주기억 장치에서 캐시 메모리로 데이터를 적재하는 기법	
주소 매핑 이유	속도 차이 감소	<ul style="list-style-type: none"> - 주 메모리는 디스크나 다른 느린 저장 장치에 비해 빠르지만, CPU에 직접 접근하는 것에 비하면 여전히 상당히 느림 - 캐시 메모리는 CPU가 자주 사용하는 데이터나 명령을 미리 읽어와 저장함으로써 이 속도 차이를 줄임
	공간/시간 지역성 활용	<ul style="list-style-type: none"> - 캐시 메모리는 이러한 지역성을 활용하여 미래에 사용될 가능성이 높은 데이터를 미리 읽어와 저장함으로써 성능을 향상
	전력 소모 감소	<ul style="list-style-type: none"> - 주 메모리에 접근하는 것은 전력을 많이 소모하는 작업 중 하나 - 따라서 캐시 메모리를 통해 더 효율적으로 작업을 수행함으로써 전력 소모 감소
	성능 향상	<ul style="list-style-type: none"> - 캐시 메모리를 적절히 사용함으로써 데이터 및 명령어에 빠르게 접근할 수 있어서 프로그램의 실행 속도를 향상

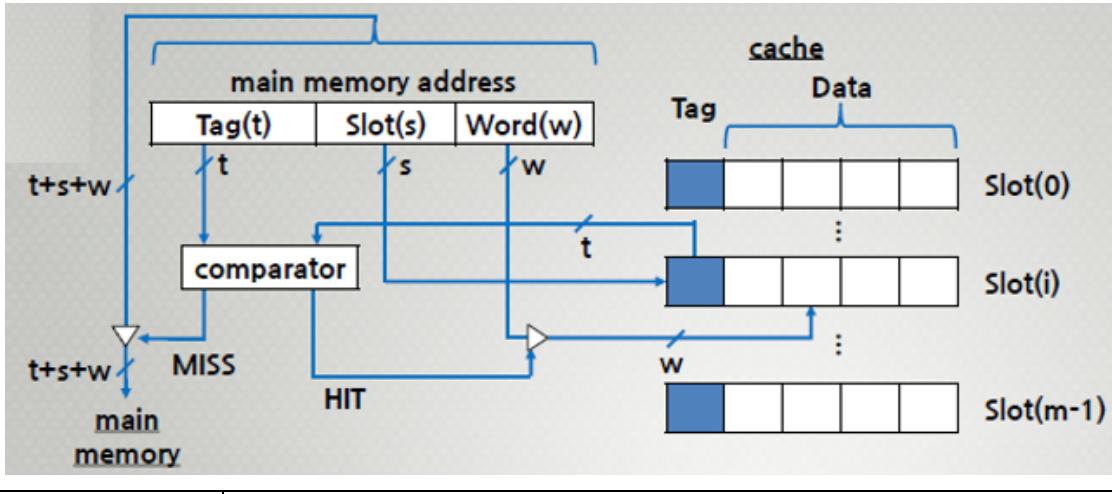
- 캐시메모리 사상 기법은 주 기억 장치의 주소 지정 방식의 의해 3가지로 분류

II. 캐시 메모리 주소 매핑 방식

가. 완전-연관 사상(Fully-Associative Mapping) 방식

구분	설명	
개요	- 주기억장치 블록이 캐시의 어떤 슬롯으로도 적재가 가능한 구조	
구성도		
동작 절차	태그 비교	- 캐시의 모든 슬롯들의 태그 영역과 내용을 비교
	캐시 적중	- 일치하는 슬롯이 있으면 캐시 적중
장점	캐시 미스	- 일치하는 슬롯이 없으면 캐시 미스
	- 새로운 블록이 캐시로 적재될 때 슬롯의 선택이 자유로움 - 지역성이 높다면 적중률이 매우 높아짐	
단점	<ul style="list-style-type: none"> - 하드웨어 구현 비용 고가, 구조 복잡 - 실제 시스템에서 거의 사용되지 않음 	

나. 직접 사상(Direct Mapping) 방식

구분	설명	
개요	- 주기억장치의 블록들이 지정된 어느 한 캐시라인으로만 적재 가능한 구조	
구성도		
동작 절차	캐시 라인 선택	- 캐시로 메모리 주소 전달 시 슬롯 비트(s)의 번호 기반 캐시 슬롯 선택
	태그 비교	- 선택 라인의 태그 비트와 주소 비교

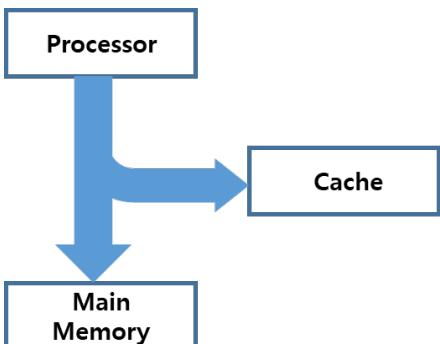
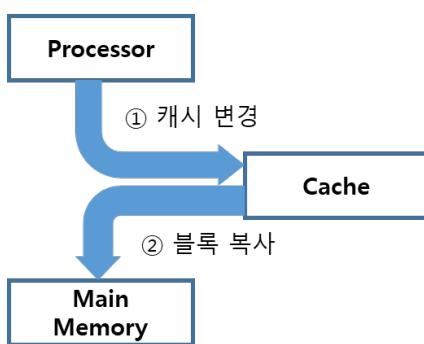
	캐시 적중 캐시 미스	- 일치 시 단어 필드 라인 내 단어 인출 - 라인 적재 후 주소 태그를 라인에 기록, 해당 라인에 다른 블록 존재 시 삭제
장점	- 간단하고, 구현하는 비용 저렴	
단점	- 라인 공유 중인 블록 적재 시 Swap-out 발생 - 실행중인 두 프로그램이 같은 슬롯을 공유하는 경우 슬롯 교체가 빈번하게 발생하여 캐시 성능 저하	

다. 세트-연관 사상(Set-Associative Mapping) 방식

구분	설명	
개요	<ul style="list-style-type: none"> 주기억장치 블록이 지정된 한 세트로만 적재될 수 있으며 각 세트는 두개 이상의 라인들로 구성된 매핑 방식 Set Associative Mapping = Direct Mapping + Fully-Associative Mapping 	
구성도		
동작 절차	세트 선택	- 메모리 주소의 세트 비트(d) 이용하여 캐시의 Set 선택
	태그 비교	- Set 내 태그와 비교
	캐시 적중 캐시 미스	<ul style="list-style-type: none"> 일치 시 캐시 적중으로 CPU로 인출 불일치 시 캐시 미스로 메모리 접근, 라인들 중 교체할 라인 결정하여 교체
세트 연관 사상의 적용	<ul style="list-style-type: none"> 2-way 세트 연관 사상이 보편적으로 사용 <ul style="list-style-type: none"> → 캐시 세트 당 두개의 슬롯을 가지는 구조 → 직접 사상보다 캐시 적중률 매우 향상 4-way 세트 연관 사상 <ul style="list-style-type: none"> → 비교적 적은 추가 비용으로 성능을 더욱 향상 	
한계점	<ul style="list-style-type: none"> 세트당 슬롯의 수를 4보다 높여도 성능은 크게 향상되지 않음 	

- 캐시 세트 번호 = (주 메모리 블록 주소) 모듈로(캐시 세트 수)

III. 캐시 쓰기 정책(Write Policy) 상세 설명

항목	Write through	Write back
개념	<ul style="list-style-type: none"> 프로세서에서 메모리에 쓰기 요청을 할 때마다 캐시의 내용과 메인 메모리의 내용을 같이 바꾸는 캐시 쓰기 정책 	<ul style="list-style-type: none"> 프로세서에서 메모리에 대한 쓰기 작업 요청 시 캐시에서만 변경 작업 후 향후 그 블록을 메인 메모리에 복사하는 캐시 쓰기 정책
구성도		
장점	<ul style="list-style-type: none"> 캐시와 메모리에 업데이트를 같이 하여, 캐시 일관성을 유지할 수 있어 안정적 	<ul style="list-style-type: none"> Write-Through 보다 훨씬 처리 속도가 빠름
단점	<ul style="list-style-type: none"> 주기억장치 또는 보조기억장치는 캐시 메모리보다 데이터 처리속도가 느리기 때문에 CPU가 대기하는 시간이 필요하여 성능이 떨어짐 	<ul style="list-style-type: none"> 속도는 빠르지만 구현하기 어렵고 캐시 메모리에만 업데이트하고 메모리에는 바로 업데이트하지 않기 때문에 캐시 일관성이 유지되기가 힘들어 종종 캐시와 메모리가 서로 값이 다른 경우 발생

- Write back 유형에서 캐시 메모리에 데이터가 Overwrite 될 때마다 새로운 Status Bit or Dirty Bit를 설정하며 실제 메모리 저장소에 데이터가 Write 되는 동작이 늦어지다 보니 Lazy Write 라고 부르기도 함

IV. 캐시 일관성 유지를 위한 기법

구분	기법	설명
SW 기법	- 공유 캐시 사용	<ul style="list-style-type: none"> 모든 프로세서가 하나의 공유 캐시 사용 항상 캐시 일관성이 유지 프로세서 간 캐시 액세스 충돌 빈번하게 발생하여 성능 저하 초래
	- 공유 변수 캐시 미사용	<ul style="list-style-type: none"> 변경 가능 공유 데이터는 주기억 장치에만 기록 캐시 저장 불가능 데이터(Non-cacheable data) : <ul style="list-style-type: none"> Lock 변수, 프로세스 큐와 같은 공유 데이터 구조 임계 영역에 의해 보호되는 데이터 캐시 적중률 저하 및 I/O 성능 저하 초래
HW 기법	- 디렉토리 프로토콜	<ul style="list-style-type: none"> 캐시 블록의 공유 상태, 노드 등을 기록하는 저장 공간인 디렉토리를 이용한 관리 기법 Full Map 디렉토리: 디렉토리에 모든 cache의 포인터와 데이터 저장

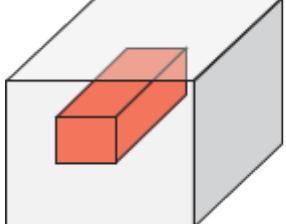
		<ul style="list-style-type: none"> - Limited 디렉토리: Full Map 디렉토리의 기억장소 부담 감소 - Chained 디렉토리: cache 포인터를 linked list로 연결, 기억장소 부담 감소
	<ul style="list-style-type: none"> - 스누피 프로토콜 (Snoopy Protocol) 	<ul style="list-style-type: none"> - 주소 버스를 항상 감시하여 캐시 상의 메모리에 접근이 있는지를 감시하는 기법 - 스누피 제어기: 다른 프로세서에 의한 메모리 액세스 감지 후 상태 조절 - 쓰기 갱신: Write 발생 시 모든 cache에 갱신된 정보 전송 - 쓰기 무효: Write 발생시 Invalid 메시지 브로드 캐스팅
프로토콜	<ul style="list-style-type: none"> - MESI 프로토콜 	<ul style="list-style-type: none"> - Modified(수정), Exclusive(배타), Shared(2개 이상의 cache에 공유), Invalid(무효, 다른 캐시가 수정) - 4가지의 상태를 통해 데이터의 유효성 여부 판단
	<ul style="list-style-type: none"> - 그 외 프로토콜 	<ul style="list-style-type: none"> - MEI, MSI, MOESI, MESIF 프로토콜 - O(Owned): 변경상태의 cache 블록을 다른 곳에서 읽은 경우 - F(Forwarding): 여로 프로세서가 공유한 cache 블록 접근 시 새로운 프로세서에게 대표로 할당

- 일반적인 SMP(Symmetric Multi Processing)/CMP(Chip-level Multi Processor) 환경에서는 캐시 일관성 지원은 필수적

“끝”

02	클라우드 정보화사업 감리 점검가이드		
문제	한국지능정보사회진흥원에서 제시한 클라우드 서비스 활용사업 감리 점검에 대하여 아래 사항을 설명하시오. 가. 공공부문의 클라우드 사업 유형 나. 클라우드 서비스 활용사업의 점검 단계 활동, 검토항목		
도메인	디지털 서비스	난이도	상 (상/중/하)
키워드	클라우드 모델, 개발 단계, 클라우드 유형		
출제배경	클라우드 서비스 활용사업 감리 개정에 따른 출제		
참고문헌	지능정보기술 감리 실무 가이드(NIA 한국지능정보사회진흥원, 2023.02)		
해설자	단합반멘토 안경환 기술사(제 110회 정보관리기술사 / akh.itpe@gmail.com)		

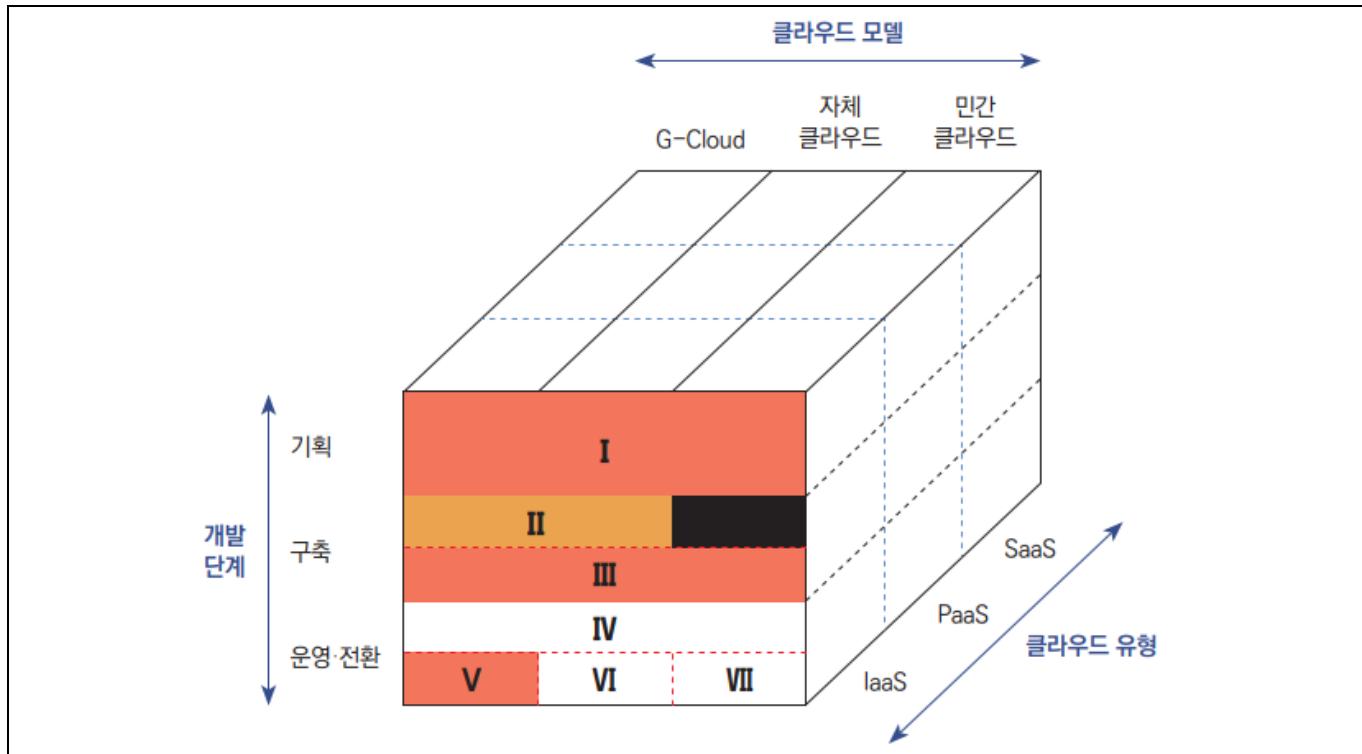
I. 공공 부문의 빠른 클라우드 전환의 밑거름. 클라우드 서비스 활용사업 감리 개요

클라우드 감리 점검가이드 개요	<p>- 최소한의 준거성을 점검할 수 있는 가이드라인 존재, 공공정보화 사업 발주현황, 그리고 위험을 인식하고 통제가능성을 기준으로 클라우드 계획수립, 클라우드 서비스활용, G-클라우드 전환 부문을 위한 점검가이드</p>		
구성	<p>개념도</p> 		사례
현행 사업 유형별 기본 점검항목에 클라우드 관련 검토항목 추가			- 클라우드 기획사업
클라우드 사업 유형을 신규로 정의하고 이에 대한 검토항목 제시			- 클라우드전환사업

- 기존 감리 점검항목에 검토항목을 추가하는 유형과 새롭게 점검항목을 만드는 2 가지 방안을 선택적으로 적용

II. 공공부문의 클라우드 사업 유형

가. 공공부문의 클라우드 사업 유형 구성도



- 클라우드 서비스 체계를 이용하여 공공 클라우드 정보화사업을 유형화할 경우, 클라우드 계획수립(I), 클라우드서비스구축(II), 클라우드서비스활용(III), 클라우드 운영(IV), G 클라우드전환(V), 자체클라우드전환(VI), 민간클라우드전환(VII)로 분류

나. 공공부문의 클라우드 사업 유형

클라우드 사업 유형	설명
I (클라우드계획수립)	- 공공부문 정보화계획수립(조직 내의 전략적 정보 요구를 파악하여 업무 활동과 이에 대한 자료영역을 기술하고, 현행 정보지원 정도를 평가하고, 정보시스템 개발을 위한 통합된 프레임워크를 제공하며, 이것을 구현하기 위하여 정보기술을 활용한 통합정보시스템 계획을 작성하는 체계적인 접근활동) 시 클라우드 서비스 도입을 일부 혹은 전부를 고려하는 사업
II (클라우드서비스구축)	- 인프라, 플랫폼(Platform)이나 소프트웨어 등 정보자원의 신규구축/재구축/고도화 등을 목적으로 하는 공공정보화사업 중 정보자원의 일부 또는 전부를 클라우드 형태로 직접 구축하는 사업
III (클라우드서비스활용)	- 인프라, 플랫폼(Platform)이나 소프트웨어 등 정보자원의 신규구축/재구축/고도화 등을 목적으로 하는 공공정보화사업 중 정보자원의 일부 또는 전부를 직접 구축 대신에 기 구축된 G-Cloud 혹은 자체 클라우드(Private Cloud), 민간클라우드를 활용하는 사업
IV (클라우드서비스운영)	- 국가기관 혹은 공공기관이 정보시스템의 인프라/애플리케이션 자원을 G-Cloud나 자체 클라우드(Private Cloud) 상에서 운영중에 있는 사업
V	- 국가기관 혹은 공공기관이 운영중인 정보시스템을 애플리케이션의 큰 변경 없이

(G-클라우드 전환)	인프라자원을 G 클라우드(Community Cloud)로 전환하는 사업
VI (자체클라우드 전환)	- 국가기관 혹은 공공기관이 운영중인 정보시스템의 어플리케이션의 변경 없이 인프라자원을 자체클라우드(Private Cloud)로 전환하는 사업
VII (민간클라우드 전환)	- 국가기관 혹은 공공기관이 기 구축/운영중인 정보시스템의 인프라 혹은 어플리케이션을 민간클라우드(Public Cloud) 서비스를 활용하는 사업

III. 클라우드 서비스 활용사업의 점검 단계 활동, 검토항목

점검 단계	활동	검토 항목
클라우드 서비스	- 클라우드 도입 적합성 평가	1. 대상업무가 클라우드 도입에 적합한 업무인가를 평가하였는가?
	- 기간 및 시기 식별	2. 조직차원의 클라우드 추진계획 및 프로젝트 포트폴리오를 통해 서 클라우드 기반 서비스 활용을 위한 적절한 기간 및 시기가 식 별되었는가?
	- 재무 영향도 검토	3. 클라우드 기반 서비스에 따라 조직의 전략 및 운영 차원에서 고 려해야 할 재정 및 예산 영향에 대해서 충분히 검토하였는가?
	- 조직능력 검토	4. 클라우드 기반서비스에 따라 새롭게 요구되는 기술적 능력에 대 해 조직차원에서 충분히 검토하였는가? 5. 클라우드 기반 서비스에 따른 조직 구조와 기술측면의 전략적 영 향도를 평가하고 이에 따른 적절한 역량확보 계획을 마련하였는 가?
	- 변화관리	6. 클라우드 기반 서비스에 따른 이해관계자(Stakeholder) 요구/문제 를 파악하고 이를 해결하기 위한 방안을 수립하였는가?
	- 거버넌스 검토	7. 클라우드 기반 서비스에 따른 구조, 지침, 통제가 적절한지를 보 장하기 위해서 조직의 거버넌스 모델(Governance Model)을 검토 하였는가?
		8. 클라우드 기반 서비스에 대한 조직차원의 적절한 비즈니스모델을 수립하고 검증하였는가? 9. 클라우드 기반 서비스에 대한 비즈니스모델이 적정한지 검증하였 는가?
클라우드 서비스 요건정의	- 위험평가	10. 비즈니스 모델을 활용하여 위협 및 위험 평가(TRA)를 수행하였 는가?
	- 요구사항 파악	11. 클라우드기반 서비스에 대한 요구사항을 파악하였는가?
	- 비즈니스사례 구축	12. 비즈니스 문제 해결과 관련된 비즈니스의 이론적 근거, 혜택, 비 용, 위험 및 옵션에 대한 객관적인 뷰(view)를 제공하기 위한 비즈 니스사례가 구축되었는가?
	- 업무연속성	13. 업무 연속성을 유지하기 위한 업무연속성 계획은 적절하게 마련 되었는가?
	- 출구전략의 준비	14. 비즈니스 연속성을 유지하면서, 다른 솔루션, 클라우드로 안전하 게 기록을 이관하기 위한 기관의 비상 계획(출구전략)이 마련되었

		는가?
	- 계약 조건의 결정	15. 계약조건은 비즈니스 요구사항, 보안요구사항 등을 만족하고 클라우드 솔루션과 관련된 발생가능 위험을 명시하였는가?
	- 클라우드 모델 결정	16. 비즈니스 문제를 해결하기 위한 최적의 클라우드 모델을 결정하였는가?
	- 서비스 제공자의 선정	17. 클라우드서비스제공자(CSP, MSP)의 제안 내용이 비즈니스의 요구 및 보안 요구 사항을 만족하고 제안 가격은 적정한지를 검토하였는가?
	- 구현계획 및 운영 계획수립	18. 인프라 및 관리 조직의 변화를 준비하기 위해 충분한 자원을 확보 등 구현계획이 마련되었는가?
클라우드 서비스 전환	- 클라우드 서비스전환 후 검토	19. 조직은 구현 후 사후 구현 검토를 수행하였는가?
클라우드 서비스 운영	- 운영관리	20. 변경관리, 장애관리, 보안관리 등 효율적인 클라우드서비스 운영 관리를 위한 활동이 적절하게 수행되었는가?
	- 서비스수준 및 사용량 측정	21. 서비스수준 및 사용량 측정은 적절하게 이루어졌는가?

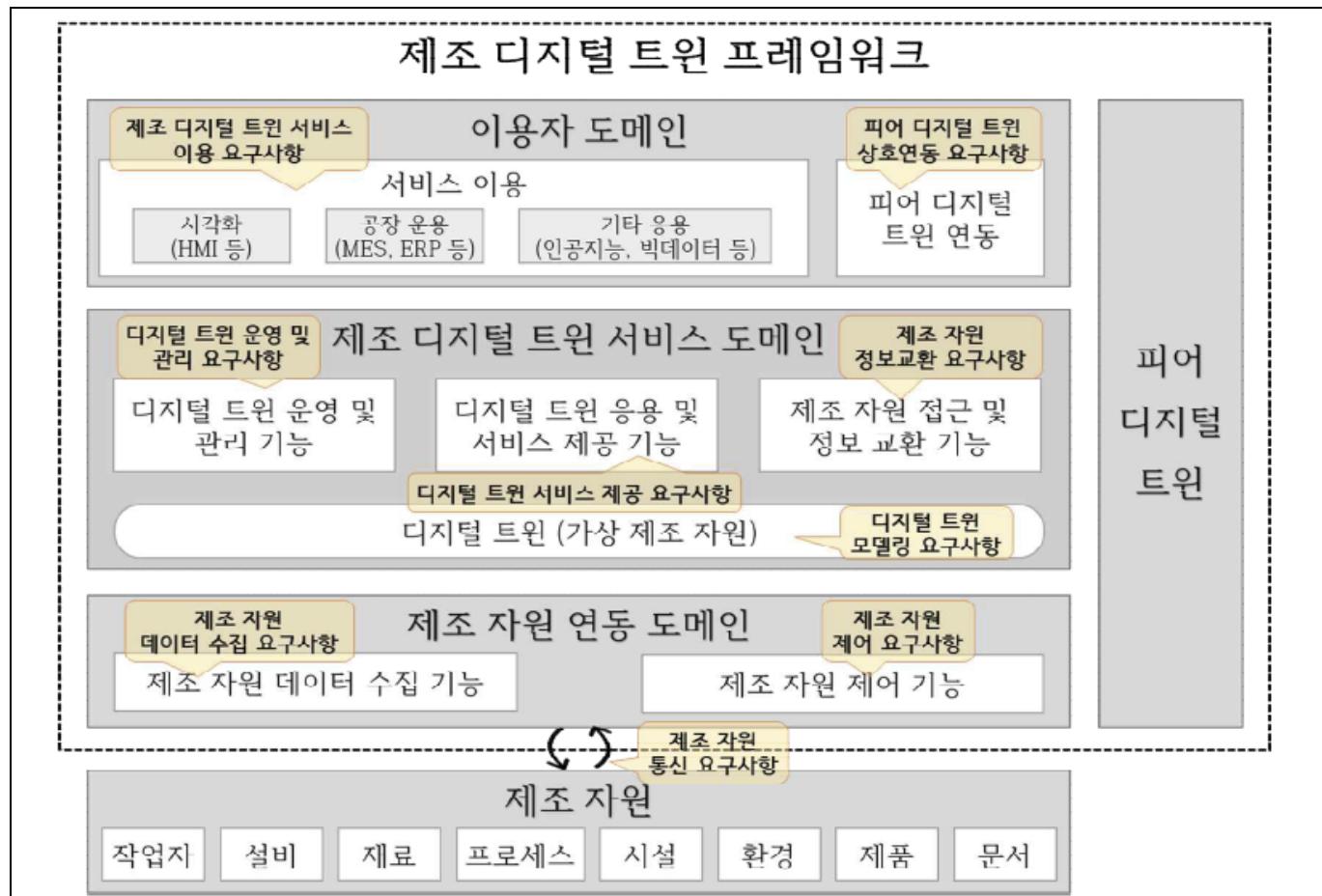
"끝"

03	디지털 트윈		
문제	<p>디지털 트윈 기술을 이용한 제조(정보통신단체표준, TTAK.KO-11.0284)에 대하여 아래 사항을 설명하시오.</p> <p>가. 제조를 위한 디지털 트윈(digital twin) 기술 개념 나. 제조 디지털 트윈(digital twin) 프레임워크 다. 액터(actor) 정보 테이블의 항목 및 항목 값</p>		
도메인	디지털서비스	난이도	중(상/중/하)
키워드	제조 자원 도메인, 제조 자원 연동 도메인, 제조 디지털 트윈 서비스 도메인, 사용자 도메인		
출제배경	디지털 트윈 확산에 따른 표준 확인		
참고문헌	정보통신단체표준, TTAK.KO-11.0284		
해설자	모멘텀 안수현 기술사(제119회 정보관리기술사 / tino1999@naver.com)		

I. 제조를 위한 디지털 트윈(digital twin) 기술 개념

- 제조 디지털 트윈은 제조 자원을 컴퓨터상에 가상적으로 표현한 가상 제조 자원

II. 제조 디지털 트윈(digital twin) 프레임워크



- 제조 디지털 트윈 프레임워크는 제조 자원 도메인, 제조 자원 연동 도메인, 제조 디지털 트윈 서비스 도메인, 사용자 도메인으로 구성되고, 이들은 각 도메인의 기능적 요구사항을 수행하는 개체인 액터(actor)로 표현

III. 액터(actor) 정보 테이블의 항목 및 항목 값

가. 제조 디지털 트윈에서의 액터의 개념

- 액터는 자신이 수행하는 기능에 따라 기기, 시스템, 애플리케이션 등의 다양한 유형으로 구분. 도메인 내의 액터들은 상호작용을 통해 도메인의 목적 달성을 지원하고, 다른 도메인 액터들과의 상호작용을 통해 다양한 서비스 제공을 가능하는 요소.
- 액터는 자신이 수행하는 기능에 따라 기기, 시스템, 애플리케이션 등의 다양한 유형으로 구분되며, 도메인 내의 액터들은 상호작용을 통해 도메인의 목적 달성을 지원하고, 다른 도메인 액터들과의 상호작용을 통해 다양한 서비스를 제공

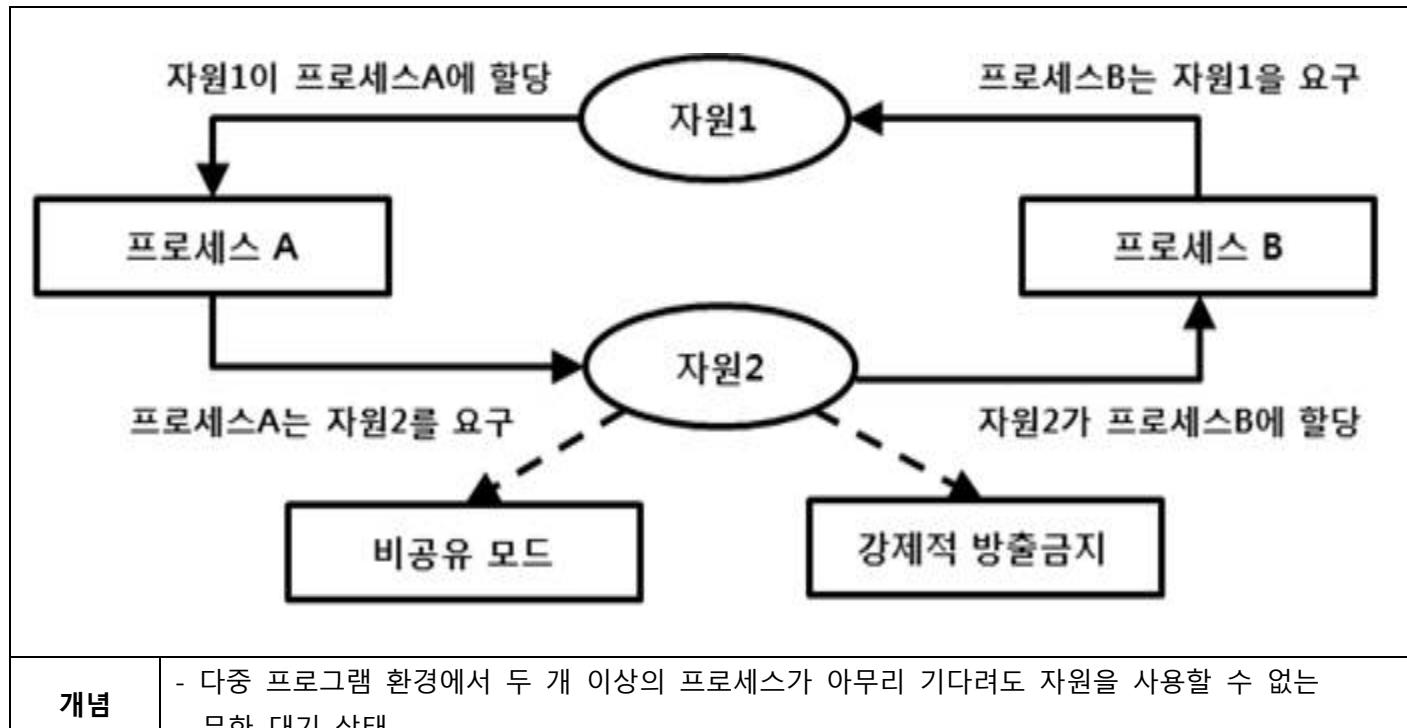
나. 액터 정보 테이블의 항목 및 항목 값

속성	속성값	설명
유형	디바이스 (device)	<ul style="list-style-type: none"> - 특정 목적을 위해 구성된 기계적, 전기적, 전자적인 기기나 장치 - 예) 공작기계
	응용 (application)	<ul style="list-style-type: none"> - 특정한 업무를 처리하는 응용 소프트웨어
	시스템 (system)	<ul style="list-style-type: none"> - 특정한 작업을 수행하는 소프트웨어와 하드웨어의 집합체
	조직 (party)	<ul style="list-style-type: none"> - 특정한 역할을 수행하는 주체로서 사람이나 조직 - 예) 작업자, 생산팀
	네트워크 (Network)	<ul style="list-style-type: none"> - 액터 등을 접속하기 위해 사용하는 선로, 교환기 등으로 구성되는 전송 매체
정보	정보 명칭	<ul style="list-style-type: none"> - 서비스 제공과 관련하여 상대 액터에 제공하는 정보의 명칭
통신 인터페이스	통신 방식	<ul style="list-style-type: none"> - 상대 액터와의 정보교환을 위해 사용하는 통신 방식 <ul style="list-style-type: none"> ※ 유선 네트워크 방식(이더넷, PLC 등) 및 무선 이동통신 방식 (ZigBee, WiFi, Bluetooth, LTE, 5G 등) ※ 단, 액터의 유형이 네트워크일 경우에는 통신 인터페이스를 별도로 기술하지 않음

“끝”

04	운영체제 DeadLock 현상		
문제	운영체제에서 발생할 수 있는 deadlock 현상에 대하여 아래 사항을 설명하시오. 가. deadlock의 개념 나. deadlock과 starvation의 차이점 다. deadlock이 발생하기 위한 조건 4가지 라. deadlock 발생 시 처리 방안		
도메인	운영체제	난이도	하(상/중/하)
키워드	상호배제, 점유와 대기, 비선점, 환형대기, 예방, 회피, 발견, 회복, 프로세스, 기아현상, 교착상태		
출제배경	운영체제의 기본 토픽에 대한 이해 점검		
참고문헌	ITPE 기술사회		
해설자	BP반 김찬일 기술사(제 130회 정보관리기술사 / s2carey@naver.com)		

I. 다중 프로세싱 환경에서 무한 자원대기, deadlock의 개념

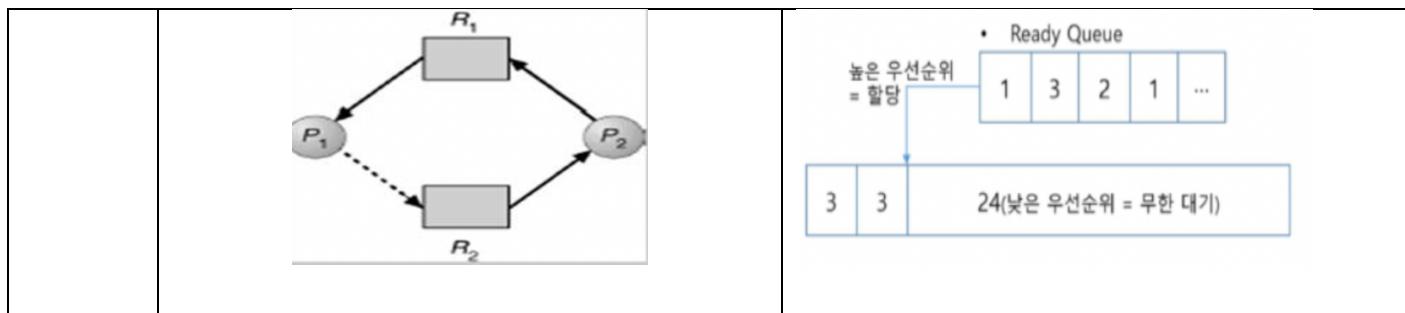


- 자원 할당 구조 문제인 deadlock 현상과, 특정 프로세스 자원 할당에 문제가 일어나는 starvation 현상 존재

II. deadlock과 starvation의 차이점

가. deadlock과 starvation의 개념적 차이점

구분	deadlock	starvation
개념	- 두개 이상의 프로세스가 서로 다른 프로세스가 보유한 자원을 기다리며 발생	- 특정 프로세스가 계속해서 자원을 할당 받지 못하고 대기하는 상태
개념도		



- deadlock 과 starvation 은 발생원인, 대기유무, 영향 범위, 해결 방법 간에 주요한 항목에서 차이점이 존재함.

나. deadlock 과 starvation 의 상세 차이점

구분	deadlock	starvation
발생 원인	<ul style="list-style-type: none"> - 자원 요청 할당이 꼬여 발생 - 상호배제, 점유대기, 비 선점, 환형대기 	<ul style="list-style-type: none"> - 특정 프로세스가 자원 얻지 못해 발생 - SJF, SRT 알고리즘
영향 범위	<ul style="list-style-type: none"> - 여러 프로세스 전체에 영향 미침 	<ul style="list-style-type: none"> - 특정 프로세스에만 영향 미침
대기 유무	<ul style="list-style-type: none"> - 모든 프로세스가 서로 대기 상태 	<ul style="list-style-type: none"> - 특정 프로세스만 대기 상태
해결 방법	<ul style="list-style-type: none"> - 예방, 회피, 발견, 복구 	<ul style="list-style-type: none"> - HRV, MLFQ 스케줄링
자원 반납	<ul style="list-style-type: none"> - 자원을 놓지 않아 해결 어려움 	<ul style="list-style-type: none"> - 자원 반납 시 해결 가능
관계 형태	<ul style="list-style-type: none"> - 다수 프로세스 상호작용 	<ul style="list-style-type: none"> - 특정 프로세스만 할당

- deadlock 은 한 시스템의 멀티 프로세싱 환경에서 다음의 4 가지 조건이 동시에 성립될 때 발생됨.

III. deadlock이 발생하기 위한 조건 4가지

가. 상호배제 및 점유와 대기 조건

조건	설명
상호배제 (Mutual Exclusion)	<ul style="list-style-type: none"> - 각 자원은 한 번에 하나의 프로세스만이 사용할 수 있어야 합니다. 즉, 한 프로세스가 자원을 사용 중이면 다른 프로세스는 해당 자원에 접근 불가능
점유와 대기 (Block & wait)	<ul style="list-style-type: none"> - 프로세스는 최소한 하나의 자원을 점유하고 있으면서 동시에 다른 프로세스에 의해 점유된 자원을 대기하는 조건 상태

- 우선순위에 의해 강제로 자원을 뺏을 수 없는 비 선점, 환형대기 조건 4 가지가 성립되어야 deadlock 발생

나. 비선점 및 환형대기 조건

조건	설명
비 선점 (Non preemption)	<ul style="list-style-type: none"> - 프로세스에 할당된 자원은 사용이 끝날 때까지 강제로 빼앗을 수 없으며, 점유하고 있는 프로세스 자신만이 해제 가능한 조건 상태
환형대기 (Circular wait)	<ul style="list-style-type: none"> - 프로세스의 집합에서 각 프로세스가 다음 프로세스가 점유한 자원을 대기하고 있어야 합니다. 이로써 자원 간에 사이클이 형성되어, 각 프로세스가 다음 프로세스가 가진 자원을 기다리고 있어야 deadlock이 발생

IV. deadlock 발생 시 처리 방안

가. 예방 및 회피 처리 방안

구분	처리 방안	내용
예방 (Prevention)	- 상호배제 조건의 부정	- 공유할 수 없는 자원을 사용할 때 성립
	- 점유와 대기 조건의 부정	- 프로세스가 자원을 요청할 때는, 다른 자원들을 점유하지 않을 것을 보장함.
	- 비 선점 조건의 부정	- 어떤 자원을 가진 프로세스가 더 이상 자원할당 요구가 받아지지 않으면 점유자원을 반납
	- 환형대기 조건의 부정	- 모든 프로세스에게 각 자원의 유형별로 할당 순서를 부여하는 방법
회피 (Avoidance)	- Wait-Die	- 자원 요청 시 대기, 점유한 자원 요청 시 종료
	- Wound-Wait	- 자원 요청 시 선점, 점유한 자원 요청 시 대기
	- Banker's Algorithm	- 자원 상태를 감시하고 사전에 자기작업에서 필요한 자원의 수를 제시하는 회피 알고리즘

- 예방 및 회피 처리방안 외에도 사전 발견 및 Kill processing 중심의 회복 기법들이 존재함.

나. 발견 및 회복 처리 방안

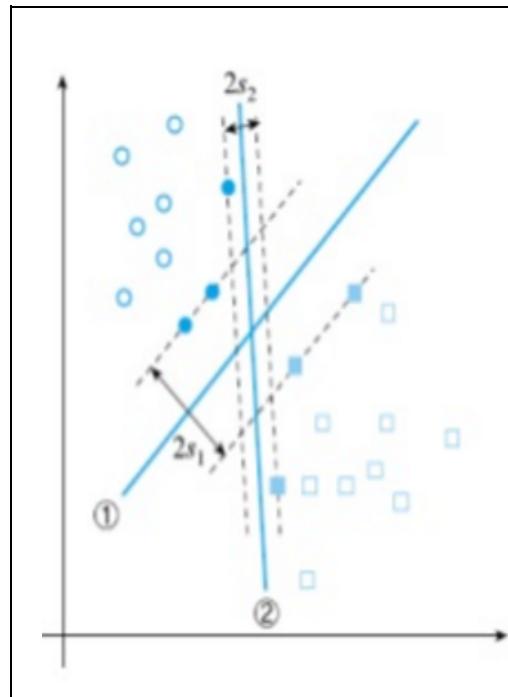
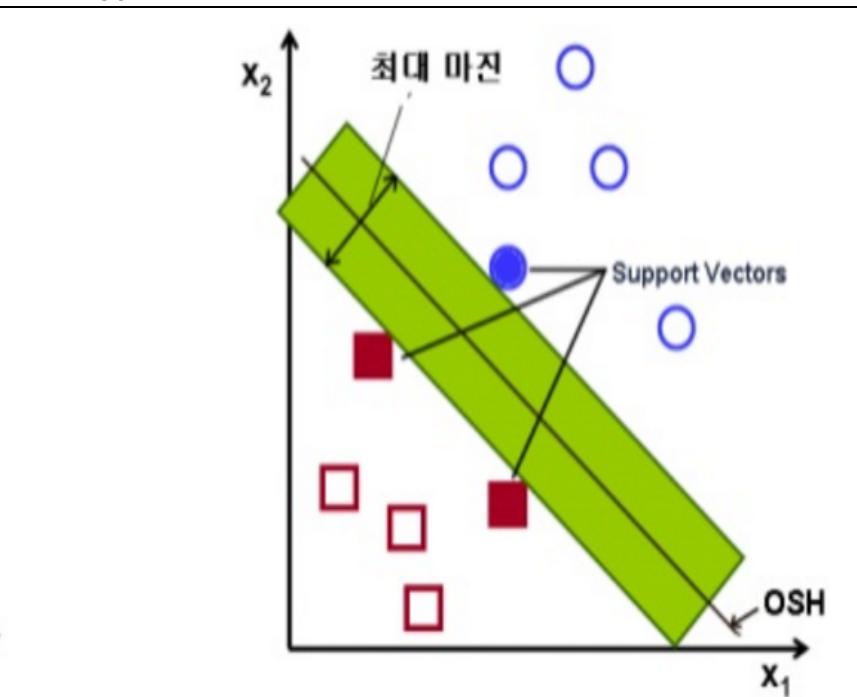
구분	처리 방안	핵심 내용
발견 (Detection)	- 자원 할당 그래프 사용	- 교착상태 발견을 목적으로 프로세스와 자원과의 관계에 따른 사이클을 나타낸 그래프
회복 (Recovery)	- 프로세스 종료	- 교착 상태 프로세스를 모두 중지하고 다시 실행
	- 자원 선점 대상 선택	- 교착 상태에 빠진 프로세스를 중에서 최소 피해를 주면서 어느 프로세스 자원 선점 할 것인지 결정

- 시스템으로부터 deadlock을 제거하여 이후로는 시스템이 deadlock 상태에 빠지지 않고 진행되게 하는 방안.

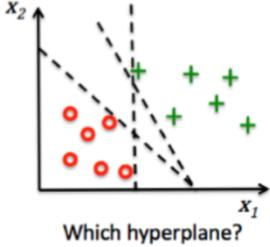
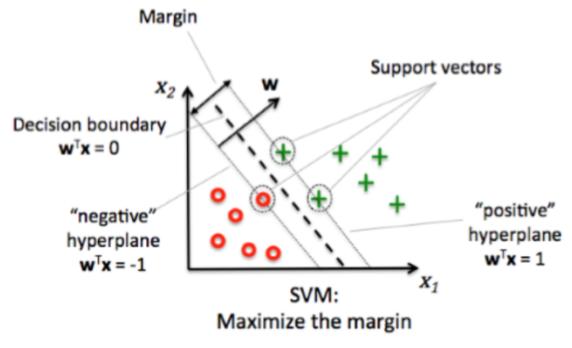
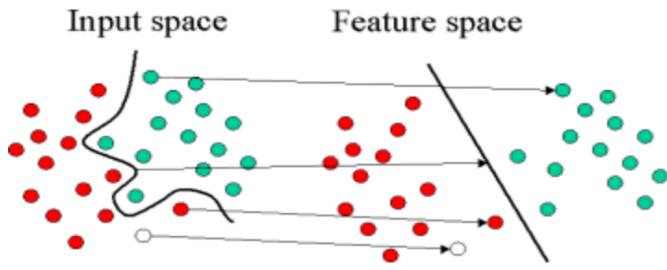
“끌”

05	SVM		
문제	서포트 벡터 머신(Support Vector Machine)은 기계학습 분야에서 많이 활용되는 학습 모델이다. 아래 사항을 설명하시오.		
	가. SVM 의 개념		
	나. SVM 의 동작방식		
	다. SVM의 장단점 및 활용사례		
도메인	인공지능	난이도	중(상/중/하)
키워드	데이터 선택, 변수 선정, 통계적 분류		
출제배경	인공지능 기본 SVM 확인		
참고문헌	ITPE 기술사회 자료집		
해설자	정상반 정상 기술사(제 124회 정보관리기술사 / itpe_peak@naver.com)		

I. 여백을 최대화하는 통계적 분석 기법, SVM(Support Vector Machine) 개념

	
<p>정의</p> <ul style="list-style-type: none"> 데이터가 사상 된 공간에서 경계선과 가장 근접한 데이터(Support Vector)간의 거리가 가장 큰 경계를 식별하는 알고리즘(MMH: Maximum Margin Hyperplane) 여백(Margin)을 최대화하여 일반화 능력의 극대화하여 결정 직선을 찾는 두 범주를 갖는 객체들을 분류하는 방법 	
<p>목적</p> <ul style="list-style-type: none"> 주어진 많은 데이터들을 가능한 멀리 두 개의 집단으로 분리시키는 최적의 초평면(hyperplane)을 Training data에서 두 클래스의 어떤 Point로부터도 최대한 멀리 떨어져 있는 Decision boundary 확보 	

II. SVM 동작 방식

구분	핵심 기술	설명
구성 요소	Support Vector	- 학습 데이터 중 분류 경계에 가장 가까운 곳에 위치한 데이터
	Margin	- 학습 데이터 중 분류 경계에 가장 가까운 데이터와 분류 경계 간 거리
	초평면 (hyperplane)	 
	커널기법 (Kernel trick)	 <ul style="list-style-type: none"> - 비선형 패턴 분리 위해 비 선형 패턴의 input space를 선형 패턴의 feature space로 변환(고차 공간)하고 해당 비선형 경계면을 찾는 방법
수행 절차	데이터 전처리	<ul style="list-style-type: none"> - 데이터를 원하는 형태로 가공하는 과정
	변수 선택	<ul style="list-style-type: none"> - 중요한 변수(독립변수)를 찾는 과정 - 너무 많은 독립변수는 정확도에 악영향 - 적절한 변수 활용 통한 정확도, 계산속도, 모형 해석 능력 향상
	통계적 분류	<ul style="list-style-type: none"> - 종속변수 예측 및 종속변수에 영향을 미치는 독립변수를 찾는 과정

- 기존 분류기는 오류율을 최소화하는 방법인 반면, SVM은 여백 Margin을 최대화하여 일반화 능력을 극대화

III. SVM의 장단점 및 활용사례

가. SVM의 장단점

구분	설명
장점	- 높은 분류 정확도(마진 최대화하며 오류 줄임)
	- 범주나 수치 예측 문제에 대하여 사용 가능
	- 노이즈 데이터에 영향을 크게 받지 않고 과적합(Over fitting) 되지 않음
	- 특히 잘 지원되는 일부 SVM 알고리즘 때문에 신경망보다 사용하기 쉬움
단점	- 최적의 모델을 찾기 위해 커널과 모델에서 매개변수의 여러가지 조합 테스트가 필요
	- 커널 함수의 선택에 따라 성능에 절대적 차이가 있음
	- 특히 입력 데이터셋이 예제 개수와 속성의 수가 많다면 훈련이 느릴 수 있음
	- 해석하기 불가능하지 않지만 어렵고 복잡한 블랙박스를 만듦

- SVM 장단점을 기반으로 다양한 영역에서 활용 진행

나. SVM의 활용 사례

사례	설명
텍스트 분류	- 텍스트와 하이퍼텍스트를 분류하여 학습 데이터를 상당히 줄일 수 있음
이미지 분류	- 사진, 이미지의 특성을 통한 분류, - 기존의 쿼리 개량 구조보다 상당히 높은 검색 정확도 확보
의료 정보 분류	- 분류된 화합물에서 단백질을 90%까지 구분
문자 인식	- 손글씨의 특징을 인지(필기체)
스팸메일 분류	- 메일함에서 메일 분류시 SVM 활용

- R에서는 LibSVM, SVM-Light 라이브러리 등을 이용하여 SVM의 활용이 가능

“끝”

06	모바일 엣지 컴퓨팅(Mobile Edge Computing)		
문제	<p>초저지연 서비스와 부하분산을 위한 모바일 엣지 컴퓨팅(Mobile Edge Computing) 기술이 확대되고 있다. 아래 사항을 설명하시오.</p> <p>가. 모바일 엣지 컴퓨팅의 개념 나. 모바일 엣지 컴퓨팅의 플랫폼 구조 및 구성요소 다. 모바일 5G에서 엣지 컴퓨팅 적용 방안</p>		
도메인	디지털서비스	난이도	중(상/중/하)
키워드	Low latency, real time analysis		
출제배경	클라우드 및 5G, 6G 인프라 중요 기술로 기본적인 이해와 점검		
참고문헌	ITPE 서브노트		
해설자	BP반 김찬일 기술사(제 130회 정보관리기술사 / s2carey@naver.com)		

I. 실시간 데이터 처리 기술, 모바일 엣지 컴퓨팅의 개념

가. 모바일 엣지 컴퓨팅의 정의

- 모바일 엣지 컴퓨팅은 분산된 네트워크 엣지(Edge)에 위치한 컴퓨팅 리소스를 활용하여 데이터 처리와 응용프로그램 실행을 지원하는 컴퓨팅 패러다임
- 클라우드 컴퓨팅과 달리, 데이터가 생성되는 지점에서 가까운 엣지 디바이스나 서버에서 처리되어 지연 시간을 줄이고 대역폭을 절약하는 것을 목표

나. 모바일 엣지 컴퓨팅의 특징

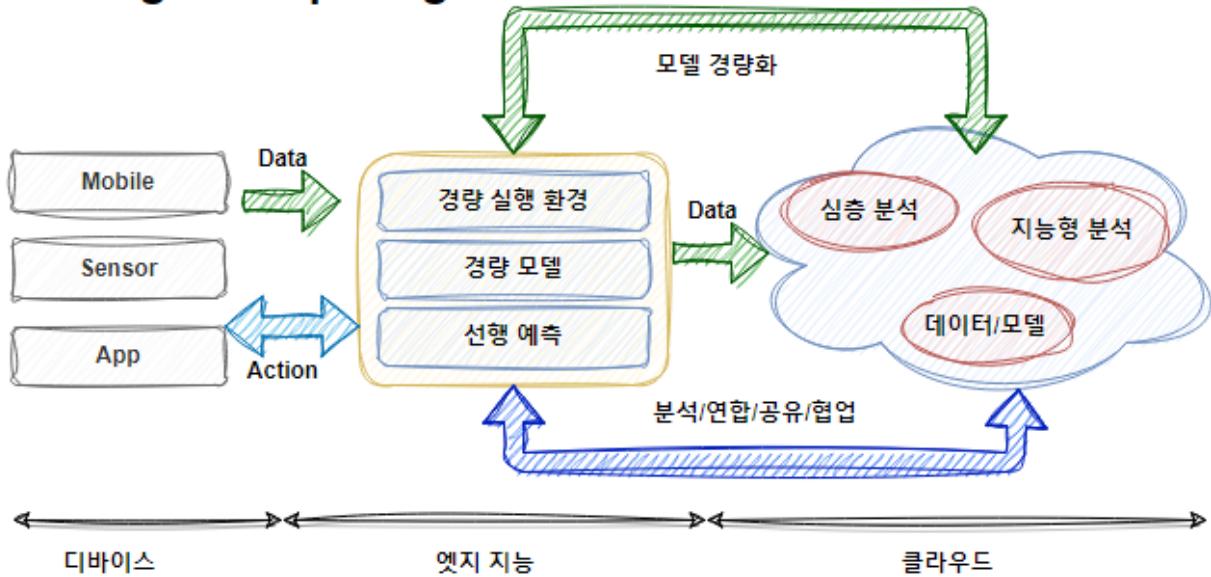
구분	특징	설명
성능 관점	- Low Latency	- 분산된 엣지 노드에서 데이터 처리를 수행하므로, 데이터의 지연 시간을 최소화하여 실시간 응답을 제공
	- 고성능 컴퓨팅 리소스	- 엣지 노드와 엣지 서버는 높은 처리 성능을 갖추어 복잡한 작업 및 응용프로그램을 효과적으로 수행
네트워크	- 대역폭 효율화	- 필요한 데이터만 중앙 데이터 센터로 전송하고, 로컬에서 데이터 처리를 수행하므로 대역폭을 효율적으로 활용
	- 데이터 로컬 처리	- 주요 데이터 처리가 엣지에서 이루어지므로 전체 네트워크에서의 데이터 트래픽이 감소
구조	- 분산된 구조	- 분산된 엣지 노드로 구성되어 있어, 특정 노드나 서버의 장애가 전체 시스템에 큰 영향을 미치지 않고 안정성을 확보

- 민감한 정보를 로컬에서 처리함으로써 중앙 데이터 센터까지의 전송 시 발생할 수 있는 보안 문제를 완화

II. 모바일 엣지 컴퓨팅의 플랫폼 구조 및 구성요소

가. 모바일 엣지 컴퓨팅의 플랫폼 구조

Edge Computing



- 주로 IoT 기기, 스마트폰, 태블릿 등의 모바일 기기와 관련이 있으며, 실시간 응용프로그램과 서비스에 적합

나. 모바일 엣지 컴퓨팅의 플랫폼 구성요소

구분	핵심 기술	설명
엣지 장치 및 인프라	- 사물 인터넷(IoT) 장치	- 데이터를 수집하고 전송하는 센서 및 액추에이터가 장착된 물리적 장치
	- 엣지 게이트웨이	- IoT 장치를 에지 네트워크에 연결하여 데이터 집계 및 제어를 가능하게 하는 장치.
	- 엣지 서버	- 데이터 처리 및 저장을 위해 네트워크 가장자리에 위치한 로컬 서버
	- 모바일 장치	- 컴퓨팅 기능을 갖춘 스마트폰, 태블릿 및 기타 휴대용 장치.
	- 산업 제어 시스템	- 산업 프로세스의 실시간 모니터링 및 제어를 위한 임베디드 시스템.
엣지 컴퓨팅 기술	- 엣지 분석	- 에지 장치 또는 에지 서버에서 로컬로 데이터를 분석하여 클라우드로의 데이터 전송 감소
	- 엣지에서 기계 학습	- 에지 장치에서 ML 알고리즘을 실행하여 실시간 추론 및 의사 결정
	- AI 추론	- 신속한 처리 및 대응을 위해 엣지에서 AI 모델 추론을 수행
	- 데이터 필터링 및 전처리	- 관련 정보를 클라우드로 보내기 전에 엣지에서 데이터를 사전 처리
	- 에지 기반 보안	- 위협으로부터 장치와 데이터를 보호하기 위해 엣지에서 보안

		조치 구현
클라우드 통합	- 중앙 집중식 클라우드 서버	- 데이터 센터에 위치한 원격 서버는 중앙 집중식 스토리지 및 컴퓨팅 성능 제공
	- 클라우드 기반 분석	- 클라우드에서 데이터를 분석하고 처리하여 집계된 데이터에서 귀중한 인사이트 도출
	- 데이터 저장	- 클라우드 기반 데이터베이스에 대용량 데이터를 저장하여 확장성과 안정성 보장
	- 데이터 동기화	- 에지 장치와 중앙 클라우드 간의 데이터 일관성 및 무결성을 보장합니다.
	- 클라우드 기반 AI 교육	- 클라우드에서 사용할 수 있는 방대한 컴퓨팅 리소스를 사용하여 복잡한 AI 모델을 교육합니다.

- 효율적인 실시간 데이터 처리, 분석 및 의사 결정을 가능하게 하며 고급 분석 위한 클라우드 리소스로 보완

III. 모바일 5G에서 엣지 컴퓨팅 적용 방안

가. 인프라와 네트워크 측면의 적용 방안

구분	적용 방안	설명
인프라	- 5G 네트워크 최적화	- 5G 기술을 활용하여 모바일 네트워크 인프라를 현대화하고 고대역폭을 적용
	- 대용량 데이터 효율적 처리 적용	- IoT 장치를 에지 네트워크에 연결하여 데이터 집계 및 제어를 가능하게 하는 장치.
네트워크	- 독립적인 가상 네트워크 적용	- 5G에서 제공되는 네트워크 슬라이싱 기술을 활용하여 독립적인 가상 네트워크를 생성
	- 응용 및 서비스별 최적화	- 네트워크 슬라이싱을 통해 응용프로그램 및 서비스에 따라 네트워크 리소스를 동적으로 최적화

- 인프라 및 네트워크 측면외에도 컴퓨팅 리소스 관리, 응용 프로그램 최적화 등을 통해 성공적으로 적용가능

나. 엣지 컴퓨팅 리소스와 응용 측면 적용 방안

구분	적용 방안	설명
리소스	- 로컬 데이터 처리 강화 적용	- 엣지 노드에 컴퓨팅 리소스를 적절히 배치하여 로컬에서의 데이터 처리를 강화
	- 엣지 노드 우선 배치 적용	- 엣지 노드를 전략적으로 배치하여 데이터가 발생하는 위치에 가까운 엣지 노드를 우선적으로 구축
응용	- 대규모 IoT, 센서 데이터 처리 최적화 적용	- 규모의 IoT 디바이스와 센서 데이터를 효율적으로 처리하기 위해 데이터 압축, 분산 처리, 스트리밍 처리 기술을 활용하여 엣지에서 데이터를 신속하게 처리
	- 스마트 시티, 홈 지원	- 엣지 컴퓨팅을 활용하여 스마트 시티 및 스마트 홈 응용프로그램에 특화된 서비스를 제공

“끝”



ITPE

ICT 온라인, 오프라인 융합 No 1

PMP 자격증 정보관리기술사/컴퓨터시스템응용기술사
IT전문가과정 정보시스템감리사
정보통신기술사 애자일

오프라인 명품 강의

ITPE 기술사회

제132회 정보처리기술사 기출문제 해설집

대상 정보관리기술사, 컴퓨터시스템응용기술사, 정보통신기술사, 정보시스템감리사 시험

발행일 2024년 01월 27일

집필 강정배PE, 안경환PE, 전일PE, 정상PE, 김찬일PE, 안수현PE

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.