



제 138 회 대비 ITPE Final Round 해설집 (3일차)

2026.02.02



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

ITPE 제 138 회 대비 Final Round

[3 일차] - 2026. 02. 02(월)

제 1 교시(시험시간: 100 분)

| 분 야 | 정보통신 | 자격 종목 | 정보관리, 컴퓨터시스템응용 | 수검 번호 | | 성 명 | |
|--------|------|----------|-------------------|----------|--|--------|--|
|--------|------|----------|-------------------|----------|--|--------|--|

※ 다음 문제 중 10 문제를 선택하여 설명 하십시오. (각 10 점)

1. 격자기반 암호화
2. 인공지능 테스트 타임 증강(Test Time Augmentation, TTA)
3. 유즈케이스 다이어그램과 클래스 다이어그램
4. 양자 머신러닝(QML, Quantum Machine Learning)
5. PoC(Proof of Concept)와 Pilot Test 비교
6. Apriori 알고리즘과 FP-Growth 알고리즘
7. 5G NSA(Non-Standalone)와 5G SA(Standalone) 비교
8. UPS(Uninterruptible Power Supply)의 주요기능과 구성요소
9. RLHF(Reinforcement Learning from Human Feedback)와 DPO(Direct Preference Optimization)
10. 내부 정렬(Internal Sorting)과 외부 정렬(External Sorting)
11. 설계검리와 종료검리의 주요점검 사항
12. CRUD 매트릭스의 구성요소와 작성 절차

[정보관리기술사 선택문제]

13. 성과경제 (Outcome Economy)의 개념과 전통적 경제와의 차이점

[컴퓨터시스템응용기술사 선택문제]

13. 운영체제에서 PCB(Process Control Block) 및 TCB(Thread Control Block)가 스케줄링에 활용되는 방식

| | | | |
|------|---|-----|-----------|
| 01 | 양자내성암호 | | |
| 문제 | 격자기반 암호화 | | |
| 도메인 | 보안 | 난이도 | 중 (상/중/하) |
| 키워드 | 최단 벡터 문제, 근접 벡터 문제, 오류가 섞인 학습 문제, 짧은 정수 해 문제 | | |
| 출제배경 | 표준이 '연구' 단계를 넘어 '공식 규격(FIPS)'으로 확정, 지속적인 양자내성암호 출제 그중 투자 및 자주 언급되고 있는 격자기반 암호화 1교시형, 2교시형 숙지 필요 | | |
| 참고문헌 | https://www.dbpia.co.kr/journal/articleDetail?nodeId=NODE11216109 | | |
| 출제자 | BP반 찬일 기술사(제 130회 정보관리, 131 컴퓨터시스템응용기술사 / s2carey@naver.com) | | |

I. 양자 내성 암호 PQC, 격자 기반 암호화 개요

가. 격자 기반 암호화 정의

- 격자 기반 암호화는 정수 격자에서의 계산적으로 어려운 문제(예: SVP-CVP, LWE-SIS)를 안전성의 근거로 하는 양자내성(Post-Quantum) 암호 기술군

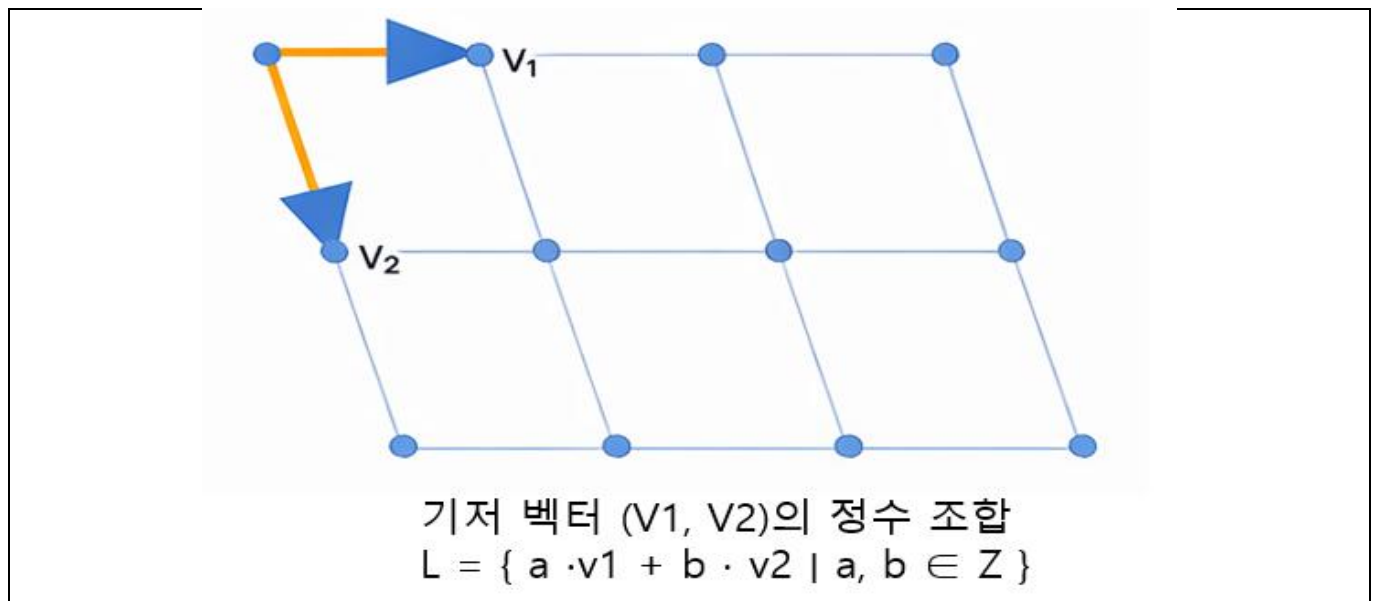
나. 격자기반 암호화 특징

| | |
|-------------|--|
| 양자 내성 | - LWE/RLWE/Module-LWE 등 격자 난제에 기반하여 양자 내성 암호로 언급 |
| 노이즈 기반 안전성 | - 선형식에 **작은 에러(노이즈)**를 섞어 역추적이 어렵게 만들어 보안 안전성 |
| 연산이 빠름 | - 다항식 연산-NTT 최적화로 소프트웨어/하드웨어 구현이 용이한 편 |
| 키·암호문 크기 증가 | - RSA/ECC 대비 통신량 / 저장 오버헤드가 커지는 경향 |

- 수학적 안전성과 별개로 타이밍/전력 분석, 오류 주입 등에 취약, 상수시간 구현, 마스킹, 검증 로직 핵심

II. 격자기반 암호화 개념도와 상세요소

가. 격자기반 암호화 개념도



- 어려운 문제들 SVP(최단 벡터 문제, CVP(최근접 벡터 문제)을 통해 양자 내성 암호화 근거를 제공함.

나. 격자기반 암호화 상세요소

| 구분 | 핵심요소 | 설명 |
|------------------|--|---|
| 난제 (보안 근거) | - Shortest Vector Problem (SVP) 최단 벡터 문제 | - 주어진 격자에서 가장 짧은 벡터를 찾는 문제로 계산적으로 매우 어려워 보안의 기반이 됨. |
| | - Closest Vector Problem (CVP) 근접 벡터 문제 | - 임의의 점에 가장 가까운 격자 벡터를 찾는 문제. 복원/복호화의 어려움을 제공. |
| | - Learning With Errors (LWE) 오류가 섞인 학습 문제 | - 잡음이 섞인 선형식을 보고 비밀을 복원하기 어렵다는 가정. 키 캡슐화 등 실사용 스킴의 핵심 근거. |
| | - Short Integer Solution (SIS) 짧은 정수 해 문제 | - 짧은 선형 관계를 찾기 어렵다는 가정. 전자서명 설계의 핵심 근거. |
| 구조화 | - Ring-LWE | - 다항식 링 위로 LWE를 일반화하여 곱셈을 컨볼루션으로 처리, 키/캡슐 크기 및 속도를 개선. |
| | - Module-LWE) | - 보안/성능 균형이 좋아 표준 KEM/서명에 널리 채택되는 일반화 형태. |
| 매개변수 | 차원·모듈 차수 | - 값이 클수록 보안은 강해지나 연산량 증가. 표준 보안등급에 맞춘 권장값 사용. |
| | 모듈러스 | - 연산 정밀도와 NTT 사용 가능 여부를 좌우. 너무 작거나 크면 성능/오차 관리가 어려움. |
| | 잡음 분포/스케일 | - LWE의 핵심 요소. 너무 작으면 공격 위험, 너무 크면 복호 실패 증가. 표준 분포 사용 권장. |
| | 실패 확률 | - 극저확률이라도 누적되면 문제. 실패 여부가 외부에 드러나지 않도록 동일 처리 필요. |
| 기본 연산 | - 수론적 변환 | - 다항식 곱셈을 빠르게 계산하는 FFT 유사 기법. |
| | - 모듈러 곱/나눗셈 가속 | - 모듈러 연산을 빠르고 일정 시간에 수행. |
| | - 상수시간 구현 | - 데이터 의존 분기·메모리 접근을 제거해 캐시·타이밍 등 측면채널 누수를 차단. |
| 구성 스킴 | - 키 캡슐화 메커니즘(KEM) | - 수신자 공개키로 캡슐을 만들고, 수신자는 비밀키로 같은 세션키를 복원. 내부적으로 잡음 보정·무결성 확인 포함. |
| | - 선택 암호문 공격 안전 변환 | - KEM을 공격자 상호작용에도 안전하게 만드는 절차. 변조/오라클 악용을 방지. |

- 표준화가 진행/완료되어 전환 이슈가 현실화: (예: ML-KEM, ML-DSA 등) → 하이브리드 전환(기존+PQC 병행)

III. 격자 기반 암호화와 아이소제니 기반 암호 비교

| 비교 항목 | 격자기반 암호 | 아이소제니 기반 암호 |
|----------|----------------------------|---------------------------------|
| 기반 난제 | - LWE/Module-LWE- 격자 난제 기반 | - 초특이 곡선 아이소제니 - 아이소제니 난제 기반 |
| 현황/주류 여부 | - KEM·서명 모두 주류- 실무 적용 활발 | - KEM은 신뢰도 하락(붕괴 사례) |
| 표준화/도입 | - 표준 확정·도입 진행- 생태계 큼 | - 평가/연구 단계- 확정 표준은 제한적 |
| 성능/구현 난도 | - 빠름 (최적화 용이), 구현 레퍼런스 많음 | - 느리거나 복잡한 편, 구현 난도 높음 |
| 크기 | - ECC 대비 큰 편- 통신·저장 부담 | - 매우 작게 가능(강점)- 대신 성능/난도 트레이드오프 |

- 격자기반은 LWE 등 격자 난제 기반의 주류 PQC이고, 아이소제니 기반은 작은 크기를 장점으로 하나 SIKE 붕괴로 신뢰도 이슈가 있어 현재는 일부 서명 중심으로 연구·평가가 이어짐.

“끝”

| | | | |
|------|--|-----|-----------|
| 02 | 인공지능 테스트 타임 증강(Test Time Augmentation, TTA) | | |
| 문제 | 인공지능 테스트 타임 증강(Test Time Augmentation, TTA) | | |
| 도메인 | 인공지능 | 난이도 | 상 (상/중/하) |
| 키워드 | TTA, 추론 단계 증강, 다중 변환 추론, 앙상블(집계), 강건성, 불확실성 | | |
| 출제배경 | 학습 단계 증강과 달리, 추론 시 입력을 여러 변환으로 반복 추론 후 집계하여 성능과 강건성을 높이는 기법으로 실무(리더보드/서비스)에서 널리 활용 | | |
| 참고문헌 | multi-crop / multi-scale testing, test-time augmentation(TTA) 튜토리얼 문헌 | | |
| 출제자 | BP반 찬일 기술사(제 130회 정보관리, 131 컴퓨터시스템응용기술사 / s2carey@naver.com) | | |

I. 추론 단계에서 입력 데이터에 다양한 변환, 인공지능 테스트 타임 증강 개요

가. 인공지능 테스트 타임 증강 정의

- TTA는 추론 시 입력을 여러 형태(반전/크롭/스케일 등)로 변환하여 각각 예측을 얻고, 이를 평균·다수결·가중합 등으로 통합해 최종 예측을 산출하는 기법

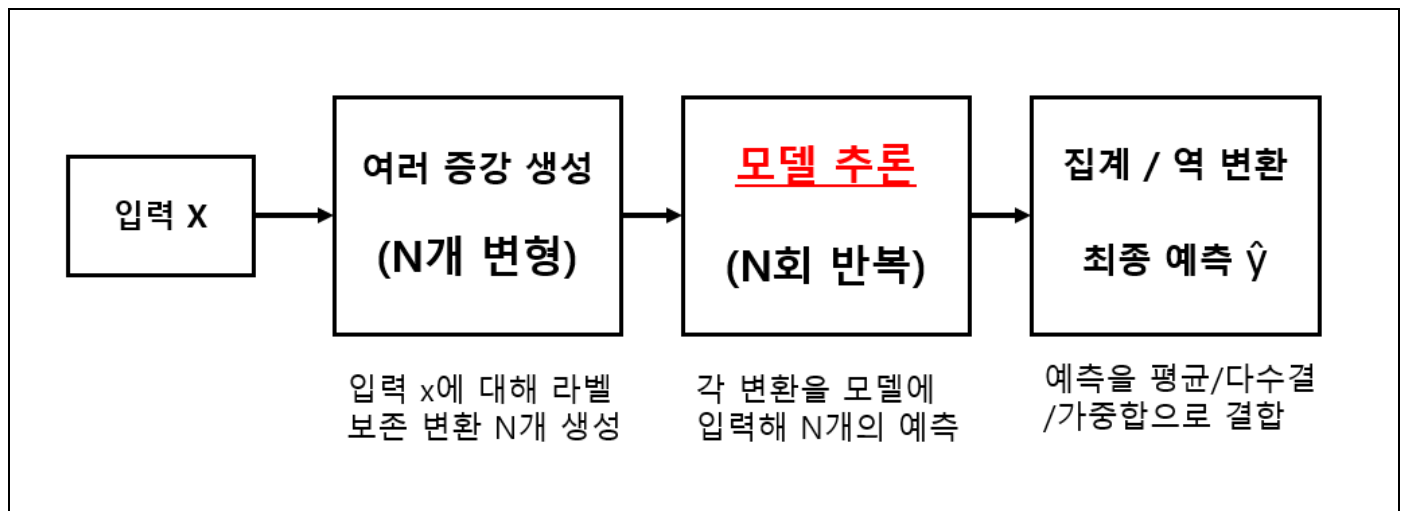
나. 인공지능 테스트 타임 증강 목적

| | |
|------------|---|
| 예측 성능 향상 | - 여러 증강 입력의 예측을 결합(평균/투표 등)해 단일 추론 대비 성능을 높임. |
| 강건성 확보 | - 회전·크롭·조명·노이즈 등 입력 변동에 덜 민감하도록 예측을 안정화. |
| 불확실성 완화 | - 다회 추론 결과를 집계해 예측 분산을 줄이고 신뢰도를 높임. |
| 실서비스 품질 개선 | - 예외 케이스에서 오분류/오탐을 줄여 운영 품질과 재현성을 개선 |

- 추론 때 여러 증강본을 돌려 예측을 합쳐 정확도와 안정성을 높이는 기법

II. 인공지능 테스트 타임 증강의 절차 및 상세 설명

가. 인공지능 테스트 타임 증강 절차



- TTA는 '추론 시 다중 변환-다중 예측-집계'로 정확도와 강건성을 높이되, 지연과 비용이 증가하는 기법

나. 인공지능 테스트 타임 증강 상세 설명

| 구분 | 핵심요소 | 설명 |
|-------|---|--|
| 입력 변환 | - flip / rotate - crop / multi-crop - scale / multi-scale | - 라벨(의미) 불변이 성립하는 변환만 선택- 동일 입력에서 N개의 증강본을 생성하여 다양한 시각 변형을 커버 |
| 추론 | - 동일 모델 반복 추론 | - 각 증강본에 대해 독립적으로 예측 수행- 출력 형태는 과제별로 상이(분류 확률/로짓, 검출 박스·점수, 분할 마스크 등) |
| 집계 | - 확률 평균, logit 평균 - 다수결, 가중합 | - 증강본별 예측을 하나의 최종 예측 \hat{y} 로 결합- 분류는 평균/투표가 일반적, 검출·분할은 정렬(역변환) 후 결합이 전제 |
| 역 변환 | - 박스/마스크 정렬 | - 증강 시 적용한 기하 변환을 출력에 역 적용하여 원본 좌표계로 복원, 검출: 박스 좌표/스케일 되돌림 |
| 불확실성 | - 분산/엔트로피 | - 변환 간 예측의 일관성(분산/엔트로피)을 통해 신뢰도 추정 가능 |

- 여러 증강본으로 반복 추론한 결과를 원본 기준으로 정렬해 합쳐 정확도 안정성과 신뢰도까지 높이는 절차

III. 학습 단계 증강(Data Augmentation)과 테스트 타임 증강(TTA) 비교

| 구분 | 학습단계 증강(Training Augmentation) | 테스트타임 증강(TTA) |
|-------|--------------------------------|---------------------------------|
| 적용 시점 | - 학습 중(데이터 로더/파이프라인) | - 추론/평가/서비스 시 |
| 목적 | - 일반화 성능 향상, 과적합 방지 | - 예측 안정화, 성능 향상, 신뢰도 개선 |
| 방식 | - 변환된 샘플로 모델 파라미터 업데이트 | - 변환 입력으로 예측을 여러 번 낸 결합 |
| 비용 | - 학습 시간 ↑(한 번), 추론 비용은 동일 | - 추론 시간 ↑(N배), 학습 비용은 동일 |
| 출력 처리 | - 보통 추가 후처리 없음 | - 검출/분할은 역 변환 후 정렬·집계 필요 |
| 효과 특성 | - 장기적으로 성능 개선 | - 즉시 효과(단, N이 커질수록 체감 감소) |

- 학습단계 증강은 모델 자체의 일반화 능력을 키우는 근본 대책이고, TTA는 추론 비용을 추가로 쓰는 대신 예측을 평균화해 즉시 안정성을 높이는 보완책

“끝”

| | | | |
|------|--|-----|-----------|
| 03 | 유스케이스 다이어그램과 클래스 다이어그램 | | |
| 문제 | 유스케이스 다이어그램과 클래스 다이어그램 | | |
| 도메인 | 소프트웨어공학 | 난이도 | 하 (상/중/하) |
| 키워드 | Use case, Actor, 시스템, 연관, 확장, 포함, 일반화, 그룹화, Public, Private, Protected, Package, 연관 관계, 일반화관계, 집합 연관관계, 복합 연관관계, 의존 관계, 실체화 관계 | | |
| 출제배경 | 137회 정보관리기술사 2교시형 최근 기출로 인해 관련 다이어그램 점검 필요. | | |
| 참고문헌 | ITPE 기술사회 자료집 | | |
| 출제자 | BP반 찬일 기술사(제 130회 정보관리, 131 컴퓨터시스템응용기술사 / s2carey@naver.com) | | |

I. 동적 UML, 유스케이스 다이어그램(Use case Diagram)의 설명

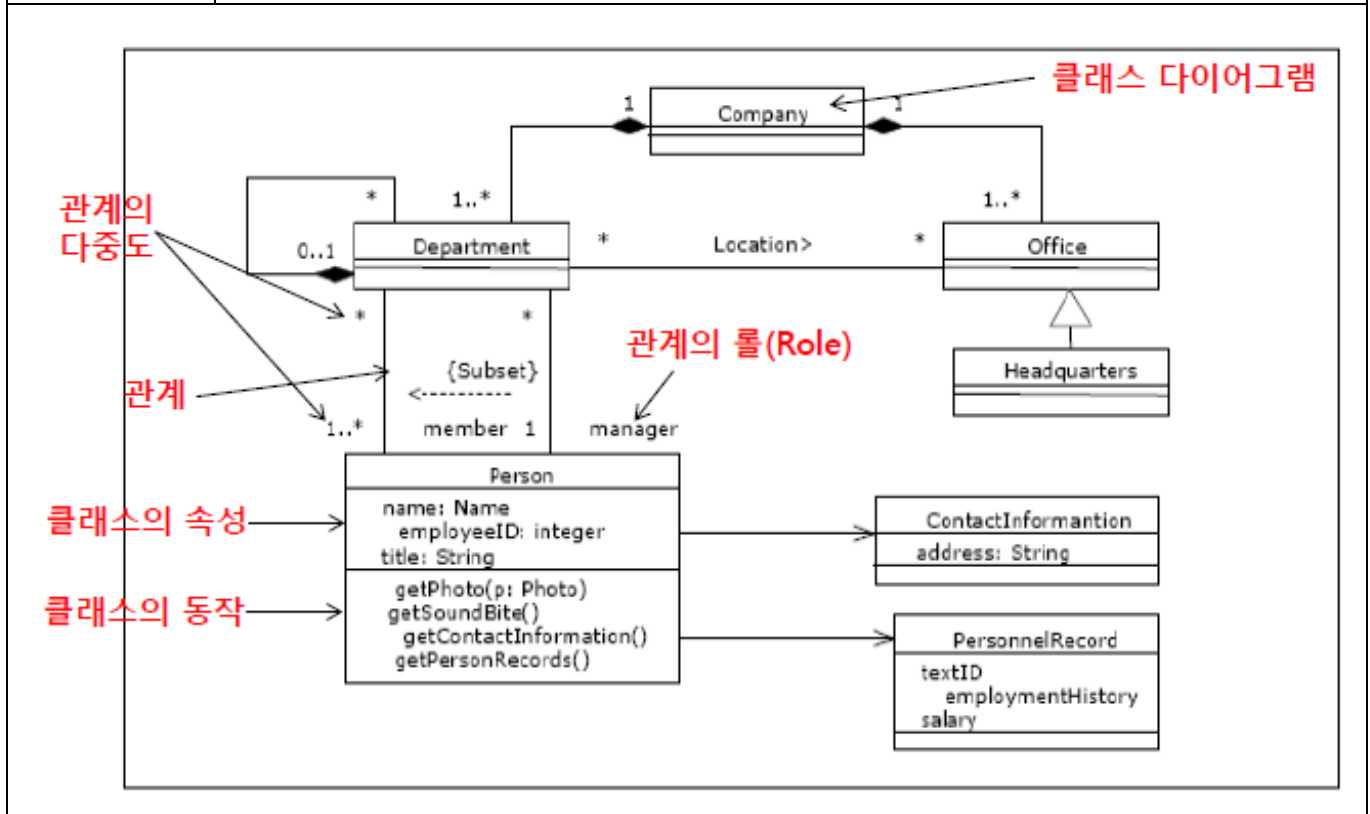
| 구분 | 설명 | |
|-------------|--|--|
| 개념 | - 시스템이 제공하고 있는 기능 및 그와 관련된 외부요소를 사용자의 관점에서 표현하는 동적 다이어그램 | |
| <div></div> | | |
| 기본 구성 | Use case | - 시스템이 제공해야 하는 서비스. - Actor가 시스템을 통한 일련의 행위 |
| | Actor (행위자) | - 사용자가 시스템에 대해 수행하는 역할(role) - 시스템과 상호작용하는 사람 또는 사물 |
| | 시스템 (System) | - 전체시스템의 영역을 표현 - 특별한 의미를 가지지 못함 |
| 관계 표현 | 연관(Association) | - Usecase와 Actor의 관계를 표현 (실선) |

| | | |
|--|-------------------------|---|
| | 확장(Extend) | - 기본 Usecase 수행 시 특별한 조건을 만족할 때 수행하는 Usecase |
| | 포함(Include) | - 시스템의 기능이 별도의 기능을 포함 (점선) - Usecase 를 수행할 때 다른 Usecase 가 반드시 수행되는 것 |
| | 일반화 (Generalization) | - 하위 Usecase/Action이 상위 Usecase/Actor에게 기능/역할을 상속 받음 |
| | 그룹화 (Grouping) | - 여러 개의 Usecase를 단순화 하는 방법 |

- 시스템 분석가가 사용자와 힘을 합쳐 시스템의 사용 방법을 결정하는데 도움을 주는 도구 동적 UML

II. 정적 UML, 클래스 다이어그램 (Class Diagram)의 설명

| 구분 | 설명 |
|----|--|
| 개념 | - 시스템을 구성하는 객체의 타입을 정의하고, 그 타입 간의 존재하는 관계를 표현하는 정적 다이어그램 |



| | | |
|-------|-----------|--|
| 접근제어자 | Public | - 어떤 클래스의 객체에서든 접근 가능 |
| | Private | - 이 클래스에서 생성된 객체들만 접근 가능 |
| | Protected | - 이 클래스의 동일 패키지에 있거나 상속 관계에 있는 하위 클래스의 객체들만 접근가능 |
| | Package | - 동일 패키지에 있는 클래스의 객체들만 접근 가능 |
| 관계 | 연관관계 | - 두 클래스들이 개념상 서로 연결 되어 있음을 의미 |
| | 일반화관계 | - 객체지향의 상속관계 개념과 동일 |
| | 집합 연관관계 | - 클래스 간의 부분과 전체의 관계를 의미 |

| | | |
|--|---------|--|
| | 복합 연관관계 | - 부분과 전체 관계이나 전체클래스 소멸시 부분클래스도 소멸하는 관계 |
| | 의존 관계 | - 한 클래스의 변화가 다른 클래스에 영향을 미치는 관계 |
| | 실체화 관계 | - 책임들의 집합인 인터페이스와 이 책임들을 실제로 실현한 클래스 사이의 관계를 나타냄 |

- 클래스 다이어그램은 정적 다이어그램으로 접근제어자와 관계로 케이스 나타냄

III. 유스케이스 다이어그램과 클래스 다이어그램 비교

| 구분 | 유즈케이스 다이어그램 | 클래스 다이어그램 |
|--------|-------------------------------------|---|
| 목적 | - 사용자 관점의 기능/요구사항 정의 | - 시스템 구조(데이터·객체·관계) 설계 |
| 관점 | - 외부(액터)-시스템 상호작용 | - 내부 클래스 간 정적 관계 |
| 주요 구성 | - Actor, Use Case, «include/extend» | - Class, Attribute/Operation, Association/Inheritance |
| 산출물 성격 | - “무엇을 한다” 중심(행위/기능) | - “무엇으로 구성된다” 중심(구조/모델) |

- 유즈케이스는 사용자 관점의 “무엇을 한다”, 클래스 다이어그램은 시스템 내부의 “무엇으로 구성된다”

“끝”

| | | | |
|-------------|--|------------|------------------|
| 04 | 양자 머신러닝(QML, Quantum Machine Learning) | | |
| 문제 | 양자 머신러닝(QML, Quantum Machine Learning) | | |
| 도메인 | 인공지능 | 난이도 | 중 (상/중/하) |
| 키워드 | 큐비트(qubit), 양자회로, 유니타리 변환, 중첩·얽힘으로 지수적 확장성, 파라미터화된 양자회로 (PQC) 최적화, 양자화학, 최적화, 금융, 양자데이터 재활용(DRU), 항등 블록(IB), 하이브리드 신경망 | | |
| 출제배경 | 137회 컴시응 기출 및 기존 레거시 인공지능 학습의 한계점을 뛰어 넘는 양자 머신러닝 주목 | | |
| 참고문헌 | ITPE 기술사회 자료집 | | |
| 출제자 | BP반 찬일 기술사(제 130회 정보관리, 131 컴퓨터시스템응용기술사 / s2carey@naver.com) | | |

I. 양자머신러닝(Quantum Machine Learning)의 개요

가. 양자머신러닝(Quantum Machine Learning)의 개념

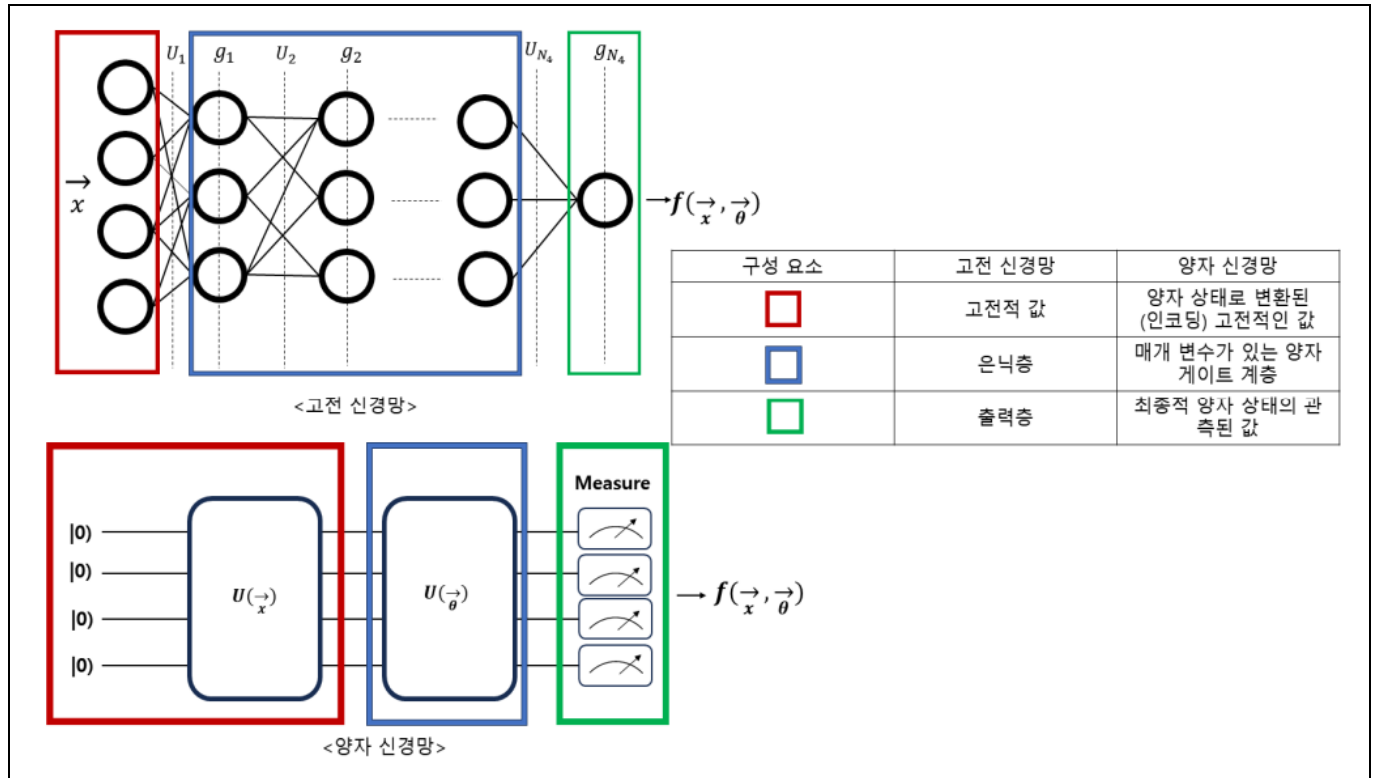
| | |
|-----------|---|
| 정의 | - 양자컴퓨팅의 원리를 머신러닝에 접목하여, 기존 고전적 머신러닝의 한계를 극복하고자 하는 첨단 융합 기술 |
| 특징 | - 양자역학의 핵심 원리인 중첩, 얽힘, 간섭을 활용하여 데이터 처리와 학습에서 새로운 가능성을 제시 |

나. 양자컴퓨팅의 핵심 원리

| 원리명 | 주요 내용 | 양자컴퓨팅에서의 역할 및 효과 |
|-----------------------|---|---|
| 중첩 (Superposition) | 큐비트가 0과 1의 상태를 동시에 가질 수 있는 양자역학적 현상 | 병렬 연산 가능, 여러 계산을 동시에 수행하여 연산 속도 극대화 |
| 얽힘 (Entanglement) | 두 개 이상의 큐비트가 서로 강하게 연결되어 한 큐비트의 상태가 다른 큐비트에 즉각 영향 | 큐비트 간 상관관계로 복잡한 문제를 효율적으로 처리, 정보 전달력 강화 |
| 간섭(Interference) | 여러 양자 상태가 서로 영향을 주어 특정 결과를 증폭하거나 감소시키는 현상 | 원하는 계산 결과의 확률을 높이고, 불필요한 결과는 억제 |

II. 양자머신러닝(Quantum Machine Learning)의 개념도 및 기존 머신러닝과 차이

가. 양자머신러닝(Quantum Machine Learning)의 개념도



- 고전 머신러닝의 구조에서 양자회로를 추가한 형태의 구조로 운영됨

나. 양자머신러닝의 기술요소 및 알고리즘

| 구분 | 사용기술 | 설명 |
|------|--|--|
| 기술요소 | - 양자 데이터 인코딩 | - 데이터를 양자 상태로 변환 - Amplitude, Angle Encoding |
| | - 파라미터화 된 양자회로 | - 학습 가능한 회로 구조 정의 - RY, RZ, CNOT 게이트 기반 회로 |
| | - 양자 커널 방법 | - 양자 상태 간 유사도 측정 - Fidelity, SWAP Test 기반 커널 |
| 알고리즘 | - QAOA(Quantum Approximate Optimization Algorithm) | - 양자 근사 최적화 알고리즘 - 조합 최적화 문제(스케줄링, 경로 탐색, 포트폴리오 선택 등)에 활용. QML에서는 분류기·강화학습 환경의 정책 최적화에도 적용 가능. |
| | - VQE(Variational Quantum Eigensolver) | - 변분적 양자 고유값 해법 - 원래는 화학/물리학에서 분자 에너지 계산을 위해 개발. 파라미터화된 양자 회로(ansatz)를 실행 → 결과를 고전 컴퓨터 최적화기로 피드백 → 반복 학습. |
| | - QNN(Quantum Neural Networks) | - 양자 신경망 - 양자 게이트를 뉴런처럼 활용하여 양자 회로 기반 신경망 |

| | | |
|--|----------------------------------|---|
| | | 을 구성. 데이터 임베딩(Quantum Embedding) → 양자 회로에서 특징 추출 → 측정 결과를 출력. |
| | - QKM(Quantum Kernel Methods) | - 양자가 데이터를 고차원 힐베르트 공간 에 임베딩 → 내적 계산(커널 함수)을 빠르게 수행. |
| | - QBM(Quantum Boltzmann Machine) | - 양자 볼츠만 머신 - 양자 중첩 상태를 활용하여 확률 분포 학습 |
| | - Data Pipeline 구축 | - Data 수집/정제/전처리/변환/축소, Data Flow 자동화 |

- QML은 고전 연산과 양자 연산의 하이브리드 아키텍처로 구성되는 경우가 많음

III. 양자머신러닝과 기존 머신러닝의 비교

| 구분 | 양자머신러닝 | 기존 머신러닝 |
|--------|----------------------|---------------|
| 데이터 단위 | 큐비트(qubit) | 비트(bit) |
| 연산 구조 | 양자회로, 유니타리 변환 | 선형 대수, 확률론 |
| 병렬성 | 중첩·얽힘으로 지수적 확장성 | 제한적 |
| 학습 방식 | 파라미터화된 양자회로(PQC) 최적화 | 역전파, 손실함수 최적화 |
| 적용 분야 | 양자화학, 최적화, 금융 등 | 이미지, 언어, 추천 등 |

- 양자의 특성을 활용하여 고속 성능이 요구되는 서비스 적용이 기대됨

“끝”

| | | | |
|------|--|-----|-----------|
| 05 | PoC(Proof of Concept)와 Pilot Test 비교 | | |
| 문제 | PoC(Proof of Concept)와 Pilot Test 비교 | | |
| 도메인 | 소프트웨어공학 | 난이도 | 중 (상/중/하) |
| 키워드 | 기술 타당성, 핵심 가설 검증, 최소 구현(프로토타입), 데모, 현장 적용, 실제/준실 운영, 사용자·업무 시나리오, KPI(효과) 검증 | | |
| 출제배경 | 자주 혼동 할 수 있는 기본 토픽 이해 점검 필요. | | |
| 참고문헌 | ITPE 기술사회 자료집 | | |
| 출제자 | BP반 찬일 기술사(제 130회 정보관리, 131 컴퓨터시스템응용기술사 / s2carey@naver.com) | | |

I. PoC(Proof of Concept)와 Pilot Test 개념 비교

| PoC | Pilot Test |
|---|---|
| <ul style="list-style-type: none"> - 아이디어나 기술이 기술적으로 성립하는지를 최소 구현으로 빠르게 검증하는 단계이며, 기술 가설을 중심으로 "될 수 있나"를 확인하는 기법 | <ul style="list-style-type: none"> - 검증된 해법을 실제 운영 환경에 적용해 사용성·업무 적합성·운영 안정성·보안·KPI 달성 여부를 점검하는 테스트 기법 |

- 즉, PoC가 "구현 가능성 확인"이라면, Pilot은 "현장 적용 가능성 및 확산 가능성 확인"

II. PoC(Proof of Concept)와 Pilot Test 상세 비교

가. PoC(Proof of Concept)와 Pilot Test 절차 비교

| 구분 | 절차 |
|-----|--|
| PoC | <div style="text-align: center;"> </div> |



- 이슈 사항과 시장의 니즈를 확인 검증 후 시장 진입에 가능성 유무를 판단하기 위해 PoC 방법 사용
- 테스트를 수행할 모집단 선정 및 테스트 서비스 시나리오의 효과적인 선택을 위해 서비스 경험이 중요

나. PoC(Proof of Concept)와 Pilot Test 상세 비교

| 구분 | PoC | Pilot Test |
|---------|------------------|------------------------|
| 단계/위치 | - 초기 검증 | - 적용 직전 검증 |
| 검증 대상 | - 기술 성립·성능·연동 | - 효과·수용성·운영성 |
| 환경 | - 개발/실험 환경 | - 실제/준실운영 환경 |
| 데이터 | - 샘플/가공 가능 | - 운영 데이터 중심 |
| 성공 기준 | - 기술 가설 입증 | - KPI 달성 + 운영 조건 충족 |
| KPI 예시 | - 지연·처리량·정확도·연동률 | - 시간절감·오류감소·만족도·장애율·비용 |
| 이해관계자 | - 개발/연구 중심 | - 현업·운영·보안 등 |
| 리스크 | - 기술 미성립 | - 운영 장애/규정/수용 이슈 |
| 변경 허용 | - 잦은 변경 가능 | - 변경 통제 필요 |
| 산출물의 형태 | - 데모·기술 결과 | - 파일럿 결과·운영안·확산안 |

- 기술이 되는지를 초기 단계에서 최소 구현으로 PoC 검증, 운영환경 적용 되는지 Pilot Test로 검증.

III. 아이디어 검증부터 상용 출시까지의 단계별 전환 흐름 제안

| 단계 | 활동 | 설명 |
|------------|------------|---|
| PoC | - 기술 검증 | - 핵심 기능을 최소 구현해 기술 성립 여부와 성능·연동 가능성을 확인 |
| Pilot Test | - 현장 적용 검증 | - 실제/준실운영 환경에서 사용자·업무 시나리오로 KPI, 운영성, 보안 요구 충족 여부를 점검합니다. |
| MVP/제품화 | - 출시 단위 정리 | - 파일럿 결과를 반영해 기능 범위를 확정하고 표준화를 통해 출시 가능한 최소 제품으로 정리 |
| UAT(인수) | - 현업 승인 | - 현업이 핵심 업무 시나리오를 기준으로 최종 검증하고 인수(승인) 여부를 확정 |
| 단계적 배포 | - 점진 출시 | - 카나리/점진 배포로 리스크를 낮추며 지표(오류율, 지연, 장애)를 모니터링해 안정성을 확인합니다. |
| 정식 출시/확산 | - 전면 운영 | - 전면 배포 후 운영 체계(SLA, 장애 대응, 개선 루프)를 정착시키고 확산을 진행합니다. |

“끝”

| | | | |
|-------------|---|------------|------------------|
| 06 | Apriori 알고리즘과 FP-Growth 알고리즘 | | |
| 문제 | Apriori 알고리즘과 FP-Growth 알고리즘 | | |
| 도메인 | 통계 | 난이도 | 중 (상/중/하) |
| 키워드 | 지지도, 신뢰도, 향상도, 하향 폐쇄성, 레벨별 탐색, 후보 생성, DB 반복 스캔, 후보 폭증, FP-Tree 압축, 후보 없음(Candidate-free), 패턴 성장(Pattern growth), 제한된 DB 스캔, 메모리/구현 복잡도 | | |
| 출제배경 | 연관규칙/빈발항목집합의 핵심 개념(support, minsup, 빈발 집합)을 정확히 이해 점검 | | |
| 참고문헌 | https://2bdbest-ds.tistory.com/entry/Apriori-Algorithm-FP-Growth | | |
| 출제자 | BP반 찬일 기술사(제 130회 정보관리, 131 컴퓨터시스템응용기술사 / s2carey@naver.com) | | |

I. 선형적 연관분석, Apriori 알고리즘 설명

| 구분 | 설명 | |
|-----------|--|---|
| 개념 | - 모든 항목 집합에 대한 지지도를 계산하는 대신, 최소 지지도 이상의 빈발 항목 집합만을 찾아 내서 연관규칙을 계산하는 기법 | |
| 절차 | <div style="display: flex; align-items: center; justify-content: space-around;"> <div style="border: 1px solid black; padding: 5px; text-align: center;">항목 빈도 계산</div> <div>→</div> <div style="border: 1px solid black; padding: 5px; text-align: center;">조합 생성</div> <div>→</div> <div style="border: 1px solid black; padding: 5px; text-align: center;">가지치기 재계산</div> <div>→</div> <div style="border: 1px solid black; padding: 5px; text-align: center;">반복 후 빈발 조합 확정</div> </div> <div style="display: flex; justify-content: space-around; margin-top: 10px;"> <div>최소 지지도 기준 미달 항목은 제거</div> <div>항목 조합 단계적 생성 (2개, 3개...)</div> <div>드문 빈도 제거 다시 빈도 계산</div> <div>종료, 결과 확장</div> </div> | |
| 특징 | 하향 폐쇄성(Apriori property) | - 부분집합이 빈발하지 않으면 상위집합도 빈발 불가 |
| | 레벨별 탐색(Level-wise) | - k-항목집합을 단계적으로 확장 |
| | 후보 생성(Candidate generation) | - 후보 집합을 만들고 지지도 계산로 필터링 |
| | DB 반복 스캔 | - 단계마다 데이터베이스를 여러 번 스캔 |
| | 후보 폭증 | - 항목이 많을수록 조합이 폭증하고, minsup이 낮을수록 더 많은 조합이 살아남아 계산량이 급증 |

- minsup : minimum support(최소 지지도)의 약자, 연관규칙/빈발항목집합 탐색에서 기준값

II. 후보 생성 없이 연관분석, FP-Growth 알고리즘 설명

| 구분 | 설명 | |
|----|---|------------------------------|
| 개념 | - 연관규칙(빈발 항목집합) 탐색에서 후보 항목집합을 일일이 만들지 않고, 거래 데이터를 FP-Tree(빈발 패턴 트리)로 압축해 빈발 패턴을 찾는 알고리즘 | |
| 절차 | <div style="display: flex; align-items: center; justify-content: space-around;"> <div style="border: 1px solid black; padding: 5px; text-align: center;">빈발 항목 선별 정렬</div> <div>→</div> <div style="border: 1px solid black; padding: 5px; text-align: center;">FP-Tree 압축 구축</div> <div>→</div> <div style="border: 1px solid black; padding: 5px; text-align: center;">조건부 경로 추출</div> <div>→</div> <div style="border: 1px solid black; padding: 5px; text-align: center;">재귀 탐색으로 추출</div> </div> <div style="display: flex; justify-content: space-around; margin-top: 10px;"> <div style="width: 22%;">최소 지지도 이상 항목 자주 나오는 순서 정렬</div> <div style="width: 22%;">공통 경로로 묶어 트리 압축 저장 항목별 연결 정보</div> <div style="width: 22%;">특정 항목이 포 함된 경로들만 모아 그 항목과 함께 나온 패턴 을 구성</div> <div style="width: 22%;">조건부 트리를 반복적으로 탐색 해 빈발 항목 조 합을 최종 추출</div> </div> | |
| 특징 | FP-Tree 압축 | - 트랜잭션을 트리로 압축해 빈도 정보를 저장 |
| | 후보 없음(Candidate-free) | - 후보 생성 없이 패턴을 직접 탐색 |
| | 패턴 성장(Pattern growth) | - 조건부 패턴 베이스/조건부 트리로 재귀 탐색 |
| | 제한된 DB 스캔 | - 보통 2회 스캔(빈도 계산 + 트리 구축) 중심 |
| | 메모리/구현 복잡도 | - 트리/조건부 트리로 메모리 부담 및 구현 난이도 |

- Apriori와 FP-Growth는 장바구니 분석 및 연관 규칙 학습에 사용되는 대표적인 데이터 마이닝 알고리즘

III. Apriori 알고리즘과 FP-Growth 알고리즘 비교

| 비교 항목 | Apriori | FP-Growth |
|--------|-----------------|------------|
| 후보 생성 | - 필요 (느림) | - 불필요 (빠름) |
| 데이터 스캔 | - 여러 번 반복 | - 두 번 |
| 핵심 구조 | - 없음 (직접 후보 검사) | - FP-Tree |
| 속도/효율 | - 상대적으로 느림 | - 빠르고 효율적 |

- FP-Growth는 Apriori의 문제점을 해결하고 대규모 데이터셋에서 연관 규칙을 찾는 데 더 적합한 알고리즘

“끝”

| | | | |
|------|---|-----|-----------|
| 07 | 5G NSA(Non-Standalone)와 5G SA(Standalone) | | |
| 문제 | 5G NSA(Non-Standalone)와 5G SA(Standalone) 비교 | | |
| 도메인 | 네트워크 | 난이도 | 중 (상/중/하) |
| 키워드 | LTE+NR, EPC, 5GC, eMBB, URLLC, mMTC | | |
| 출제배경 | 정부가 3G-LTE 주파수 재할당 조건으로 5G SA(단독망) 구축/서비스 제공을 의무화하면서, 통신 3사가 SA 전환을 본격 준비 | | |
| 참고문헌 | ITPE 서브노트 | | |
| 출제자 | 강평야 전일 기술사(제 114 회 정보관리기술사 / ji@naver.com) | | |

I. 5G의 NSA(Non-Standalone)와 SA(Standalone) 개념 비교

| 구분 | NSA | SA |
|----|--|--|
| 정의 | - 기존 4G LTE 코어망(EPC)에 5G 기지국을 결합하여 5G 서비스를 제공하는 과도기적 네트워크 구조 | - 5G 전용 코어망(5GC)과 5G 기지국으로 독립 구성된 완전한 5G 네트워크 구조 |
| 특징 | - LTE 앵커, 빠른 상용화, 제한적 5G 기능 | - 5G 전용 코어, 네트워크 슬라이싱, 초저지연 지원 |

-NSA와 SA 두가지 구조 모두에서 단말은 4G, 5G 두 무선 접속을 동시에 지원하는 형태로 진화

II. NSA와 SA 상세 비교

가. NSA와 SA 아키텍처 비교

| 구분 | NSA | SA |
|----------|----------------------------|--------------|
| 3GPP 릴리즈 | - Release 15 | - Release 15 |
| 구성도 | | |
| 무선망 | - LTE + NR(New Radio) | - NR |
| 핵심망 | - EPC(Evolved Packet Core) | - 5GC(Core) |

가. . NSA 와 SA 세부 비교

| 구분 | NSA | SA |
|-----------|------------------------|-----------------------------|
| 구축 목적 | - 빠른 5G 상용화 및 비용 절감 | - 5G 고유 기능 완전 구현 |
| 지연 시간 | - LTE 의존으로 지연시간 개선에 한계 | - 초저지연(URLLC) 지원 |
| 핵심 기능 | - eMBB 중심 (고속 데이터) | - eMBB + URLLC + mMTC |
| 네트워크 슬라이싱 | - 제한적 또는 미 지원 | - 완전 지원 |
| 확장성 | - 기존 LTE 구조에 종속 | - 자율주행, 스마트팩토리 등 산업용 서비스 최적 |
| 구축 난이도 | - 낮음 (기존 인프라 활용) | - 높음 (전면 신규 구축) |

- 정부는 5G SA 의무화를 통해 주파수 공공성 확보와 미래산업 및 공공 서비스의 경쟁력 강화를 꾀하고 있음

III. 정부의 5G SA 의무화 추진 목적과 반대 의견

| 구분 | 주요 내용 |
|--------|---|
| 의무화 이유 | - 5G 고유 기능(네트워크 슬라이싱, URLLC 등) 구현을 위해 SA 전환이 필요 |
| | - 자율주행·스마트팩토리·공공안전 등 국가 전략 산업 경쟁력 확보 |
| | - 주파수 공공자원 재할당에 따른 투자·품질 고도화에 대한 정책적 책임 부과 |
| 반대 의견 | - 초기 SA 전환 시 NSA 대비 체감 속도 저하 가능성 |
| | - 5GC 구축 및 망 고도화에 따른 통신사의 대규모 투자 부담 |
| | - 시장 수요 대비 정책 주도형 전환으로 과도한 규제라는 지적 |

- NSA-SA 병행을 통한 단계적 전환과 정책 유연화, 그리고 SA 기반 서비스 창출을 병행하여 투자 부담과 체감 품질 저하를 최소화하는 방향으로 진행 필요

“끝”

| | | | |
|-------------|---|------------|------------------|
| 08 | UPS(Uninterruptible Power Supply) | | |
| 문제 | UPS(Uninterruptible Power Supply)의 주요기능과 구성요소 | | |
| 도메인 | 디지털서비스 | 난이도 | 중 (상/중/하) |
| 키워드 | 무정전 전원공급, 전원 품질보장, 시스템 보호, 정류기, 인버터, 축전지, 동기절체 스위치 | | |
| 출제배경 | 최근 국가 핵심 정보자원 화재·정전 사고를 계기로 UPS 등 전원 인프라가 단순 설비가 아닌 국가 서비스 연속성을 좌우하는 핵심 시스템 요소임을 이해하고 있는지를 검증 | | |
| 참고문헌 | ITPE 서브노트, ITPE 모의고사 | | |
| 출제자 | 강평야 전일 기술사(제 114 회 정보관리기술사 / ji@naver.com) | | |

I. 무 정전 전원 공급장치, UPS의 주요 기능

가. UPS(Uninterruptible Power Supply) 개념

- 상용 전원에서 발생 가능한 전원 장애를 극복하여 양질의 **안정된 교류 전력을 공급하는 장치**
- 일반 전원 또는 예비 전원 등을 사용할 때 전압 변동, 주파수 변동, 순간 정전, 과도 전압 등으로 인한 전원 이상을 방지

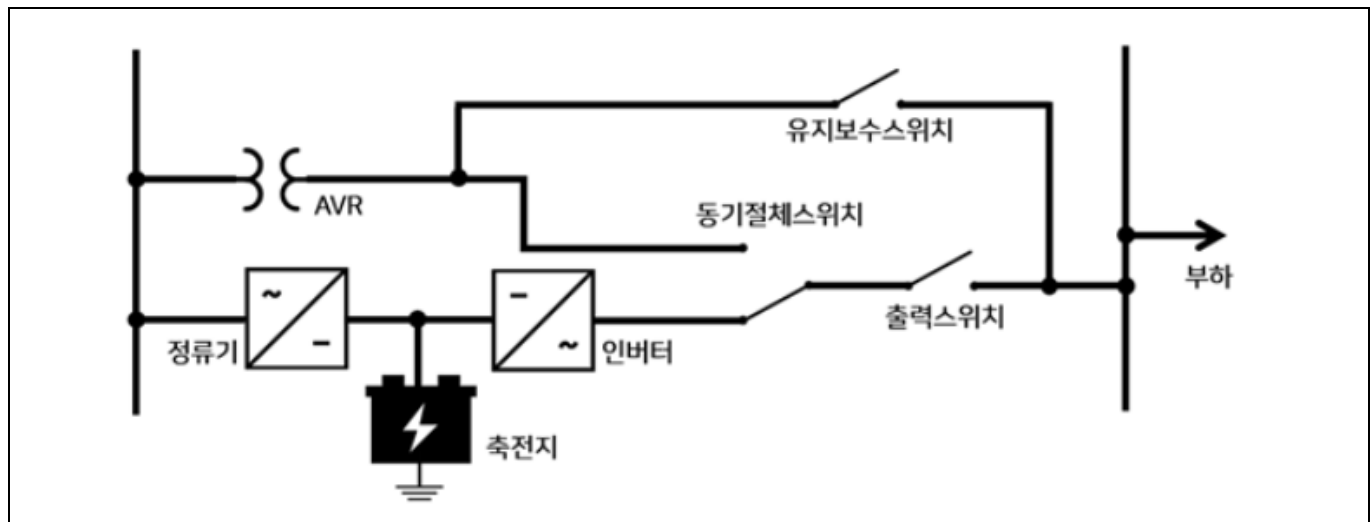
나. UPS 주요 기능

| 구분 | 주요 기능 | 설명 |
|--------|-------------------------|----------------------------------|
| 전원 안정화 | 무정전 전원공급 | - 정전·순간정전 발생 시 배터리를 통해 연속 전원 제공 |
| | 전원 품질보장 | - 전압강하, 서지, 노이즈 등 전원 품질 이상 제거 |
| 연속성 확보 | 시스템 보호 | - 전원 이상으로 인한 서버 장애·데이터 손상 방지 |
| | 장애 완충 및 장애 확산 방지 | - 상용전원-발전기 전환 완충 및 대규모 서비스 중단 차단 |

- UPS는 전원 안정화와 서비스 연속성을 동시에 책임지는 핵심 인프라

II. UPS 시스템 구성요소

가. UPS 시스템 구성도



나. UPS 시스템 구성요소

| 구성요소 | 설명 |
|--|--|
| 정류기(Rectifier) 인버터(Inverter) | - 정류기는 AC 전원을 DC로 변환 하여 축전지 및 인버터에 전력을 공급 - 인버터는 변환된 DC 전원을 다시 안정된 AC로 변환 하여 부하에 공급 |
| 축전지(Battery) | - 평상시에 정류기에서 받은 DC 전력 을 모아뒀다가 정전 시 혹은 정류기의 고장 시 부하에 전력을 공급 |
| 동기 절체 스위치 (Static Transfer Switch) | - 동기 절체 스위치는 부하에 공급하는 전원을 끊김 없이 절체 해주는 역할 - UPS 시스템 내부 고장 시 상용전원으로 Bypass 하고, 고장 복구 후에는 다시 UPS 선로 측으로 선로를 변경하는 역할 (OFF-LINE 방식에서는 반대) |
| 유지 보수 스위치 (Maintenance Bypass Switch) | - 주로 MBS 라는 약자로 많이 불리며, UPS 의 인위적인 유지 보수 및 점검이 필요할 때 상용전원에 Bypass 하는 기능 |
| AVR(Automatic Voltage Regulator) | - 자동전압조정기, 혹은 자동변압기로 불안정한 전압변동을 일정하게 출력 - 상용전원 사용 시 고품질의 전력을 부하에 입력해 주기 위한 장비 |

- UPS 는 정류기·인버터·축전지·동기절체 스위치 등으로 구성되어 상용전원 장애 시에도 전원 품질을 유지하며 무중단 전원 공급을 보장

III. UPS 시스템 설계 및 운영 적용 시 고려사항

| 구분 | 고려사항 | 설명 |
|-------|------------|-------------------------------------|
| 설계 관점 | 용량 산정 | - 시스템 중요도 및 최대 부하를 고려한 적정 UPS 용량 산정 |
| | 이중화·연계 | - N+1 구성, 발전기 및 DR 센터와의 전원 연계 설계 |
| 운영 관점 | 모니터링 ·유지보수 | - 배터리 상태 감시, 정기 점검 및 예방 정비 수행 |
| | 장애·비용 관리 | - 장애 대응 절차 수립 및 투자 대비 효과(ROI) 고려 |

- UPS 는 설계·운영 전 주기에 걸친 관리 역량을 요구하는 핵심 인프라임

“끝”

| | | | |
|------|---|-----|-----------|
| 09 | RLHF(Reinforcement Learning from Human Feedback)와 DPO(Direct Preference Optimization) | | |
| 문제 | RLHF(Reinforcement Learning from Human Feedback)와 DPO(Direct Preference Optimization) | | |
| 도메인 | 인공지능 | 난이도 | 중 (상/중/하) |
| 키워드 | 보상모델(Reward Model), 강화학습(PPO), 인간 선호 정렬(Alignment), 직접 선호 최적화, 보상모델 제거, 학습 구조 단순화 | | |
| 출제배경 | LLM 학습 패러다임의 전환 이해 확인 | | |
| 참고문헌 | ITPE 서브노트 | | |
| 출제자 | 강평야 전일 기술사(제 114 회 정보관리기술사 / ji@naver.com) | | |

I. 모델의 행동 조정, RLHF 상세 설명

가. RLHF(Reinforcement Learning from Human Feedback) 개념

| 구분 | 세부 내용 | |
|----|---|-------------------------------------|
| 정의 | - 사람의 평가를 보상 신호로 활용하여, 강화학습을 통해 AI 모델의 출력을 인간의 기대와 가치에 맞게 정렬(Alignment)하는 학습 기법 | |
| 특징 | 인간 중심 정렬(Alignment) | - 정확성 뿐 아니라 유용성, 안전성, 윤리성을 함께 반영 |
| | 보상모델 기반 강화학습 | - 보상을 기준으로 모델을 강화학습(PPO 등)으로 최적화 |
| | 고품질 출력 유도 | - 환각(Hallucination), 공격적·부적절한 응답 감소 |

나. RLHF의 처리 절차

| <div> <div> <p>① 지도 미세조정 (Supervised Fine-Tuning, SFT)</p> </div> <div> <p>② 보상모델 학습 (Training a Reward Model)</p> </div> <div> <p>③ 정책 최적화 (Optimize Policy, PPO)</p> </div> </div> | | |
|--|-------------------------|---|
| 단계 | 주요 입력 | 수행 내용 |
| 지도 미세조정 (SFT) | 인간 시연 데이터 (질문-모범 답변) | - 인간이 작성한 고품질 응답 수집 - Base LLM 에 지도학습 기반 미세조정 수행 |
| 보상모델 학습 | 인간 선호 데이터 (응답 비교 결과) | - 동일 프롬프트에 대한 복수 응답 생성 - 인간이 선호 응답 선택• 선호 데이터를 이용해 보상모델 지도학습 |

| | | |
|--------------|---------------|---|
| 정책 최적화 (PPO) | SFT 모델 + 보상모델 | <ul style="list-style-type: none"> - SFT 모델을 초기 정책으로 설정 - 보상모델로 응답 평가 - PPO 기반 강화학습으로 정책 업데이트 반복 |
|--------------|---------------|---|

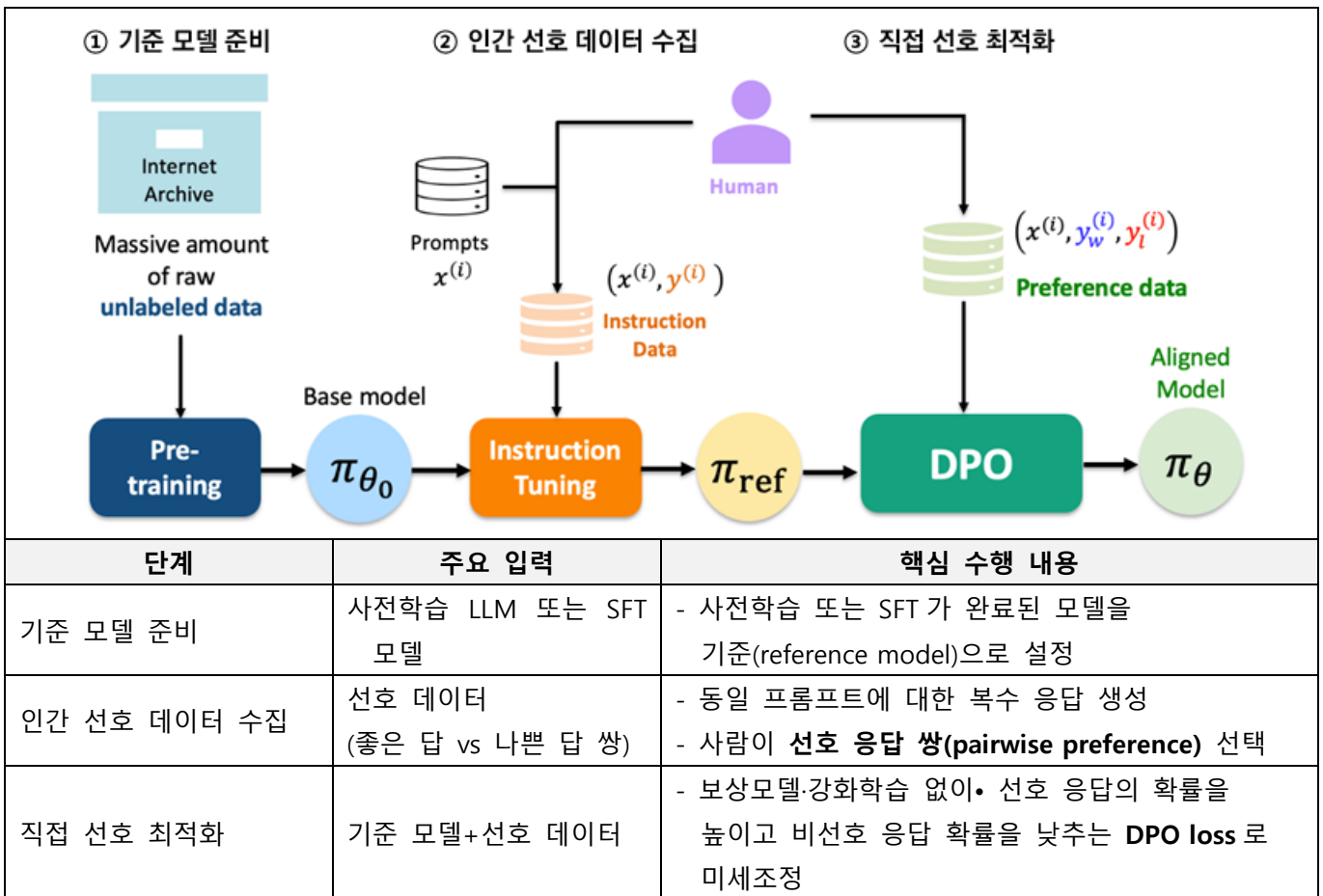
- RLHF 는 인간 시연 데이터로 SFT 모델을 생성하고, 인간 선호로 보상모델을 학습한 뒤, PPO 강화학습을 통해 정책을 최적화하는 3 단계 학습 절차

II. 모델의 행동 조정, RLHF 상세 설명

가. DPO(Direct Preference Optimization) 개념

| 구분 | 세부 내용 | |
|----|---|--|
| 정의 | - 인간의 선호 데이터를 보상모델이나 강화학습 없이, 목적함수에 직접 반영하여 언어모델을 최적화하는 학습 기법 | |
| 특징 | 보상모델·강화학습 불필요 | - 인간의 평가 결과를 간접 보상이 아닌 직접 목적함수(DPO loss)로 사용 |
| | 인간 선호의 직접 반영 | - 보상을 기준으로 모델을 강화학습(PPO 등)으로 최적화 |
| | 학습 안정성 우수 | - 강화학습 특유의 불안정성(보상 붕괴, 튜닝 난이도) 없음 |

나. RLHF 의 처리 절차



- 사람이 더 선호한 응답의 확률은 높이고, 덜 선호한 응답의 확률은 낮추는 방식으로 모델을 직접 미세조정

III. RLHF 와 DPO 비교

| 구분 | RLHF | DPO |
|--------|--------------------|-----------------|
| 학습 방식 | - 보상모델 + 강화학습(PPO) | - 선호 데이터 직접 최적화 |
| 구조 복잡도 | - 높음 | - 낮음 |
| 학습 안정성 | - 튜닝 난이도 높음 | - 안정적 |
| 운영 비용 | - 높음 | - 낮음 |
| 실무 적합성 | - 대규모 서비스 | - 기업·공공 환경 |

- RLHF 는 정교하지만 복잡하고, DPO 는 단순하면서도 실무 친화적인 대안

“끝”

| | | | |
|------|---|-----|-----------|
| 10 | 내부 정렬(Internal Sorting)과 외부 정렬(External Sorting) | | |
| 문제 | 내부 정렬(Internal Sorting)과 외부 정렬(External Sorting) | | |
| 도메인 | 알고리즘 | 난이도 | 중 (상/중/하) |
| 키워드 | 메모리 기반, CPU 연산 중심, 고속 처리, 디스크 기반, 대용량 처리, I/O 최소화 | | |
| 출제배경 | 알고리즘의 기본지식 확인 | | |
| 참고문헌 | 서브노트, ITPE 기술사회 자료 참조 | | |
| 출제자 | 강평야 전일 기술사(제 114 회 정보관리기술사 / ji@naver.com) | | |

I. 연산 효율의 핵심, 내부 정렬 상세 설명

가. 내부 정렬(Internal Sorting) 개념

| 종류 | 설명 |
|----|---|
| 개념 | - 데이터의 크기가 주 기억장소 용량보다 적을 경우 기억장소를 활용 하여 정렬하는 방법 |
| 특징 | - 메모리 기반 · 고속 처리 · CPU 연산 중심 |

나. 내부 정렬 종류

| 구분 | 알고리즘 | 주요 내용 |
|---------|------------------------|-------------------------------|
| 비교 기반 | 버블 정렬 (Bubble Sort) | - 인접 요소 교환, 구현 단순, 성능 낮음 |
| | 선택 정렬 (Selection Sort) | - 최소/최대 선택 후 교환, 교환 횟수 적음 |
| | 삽입 정렬 (Insertion Sort) | - 부분 정렬 유지, 거의 정렬된 데이터에 효율적 |
| | 병합 정렬 (Merge Sort) | - 분할정복, 안정 정렬, 추가 메모리 필요 |
| | 퀵 정렬 (Quick Sort) | - 평균 성능 우수, 불안정 정렬 |
| | 힙 정렬 (Heap Sort) | - 완전이진트리 기반, 항상 $O(n \log n)$ |
| 비 비교 기반 | 계수 정렬 (Counting Sort) | - 키 범위 제한, 매우 빠름 |
| | 기수 정렬 (Radix Sort) | - 자릿수 기준 정렬, 비교 연산 없음 |
| | 버킷 정렬 (Bucket Sort) | - 분포 기반, 균등 분포 시 효율적 |

- 내부정렬은 메모리 내에서 비교·교환 연산을 중심으로 CPU 연산 효율을 극대화하는 정렬 방식임

II. 디스크 I/O 최소화를 위한 외부 정렬 상세 설명

가. 외부 정렬(External Sorting) 개념

| 종류 | 설명 |
|----|--|
| 개념 | - 데이터의 크기가 주기억장소의 용량보다 클 경우 외부 기어 장치(디스크, 테이프 등) 를 사용하여 정렬하는 방법 |
| 특징 | - 디스크 기반 · 대용량 처리 · I/O 최소화 |

나. 외부 정렬 종류

| 그룹 | 알고리즘 | 주요 내용 |
|---------|----------|----------------------------|
| 런 생성 단계 | 대체 선택 정렬 | - 힙 기반으로 런 길이를 확장하여 I/O 감소 |
| 병합 단계 | 2-way 병합 | - 두 개의 정렬된 런을 순차 병합 |
| | k-way 병합 | - 다수 런을 동시에 병합하여 병합 횟수 감소 |
| | 다단계 병합 | - 병합 결과를 반복 병합하여 최종 정렬 완성 |

- 외부정렬은 초기 런 생성과 k-way 병합을 통해 디스크 I/O 횟수를 최소화하는 정렬 방식임

III. 기밀성과 무결성을 높이기 위한 접근 제어 방법



| 구분 | 내부정렬 | 외부정렬 |
|-----------|---------------------------------|-----------------------|
| 사용 기억장치 | - 주기억장치(RAM) | - 보조기억장치(디스크) + RAM |
| 핵심 성능 요소 | - CPU 연산 효율 | - 디스크 I/O 최소화 |
| 처리 데이터 규모 | - 소·중규모 데이터 | - 대규모 데이터 |
| 알고리즘 방식 | - 교환·선택·삽입·병합·분배·계수 | - 런 생성·병합 기반 |
| 대표 알고리즘 | - Quick, Heap, Merge, Insertion | - External Merge Sort |
| 처리 속도 | - 빠름 | - 상대적으로 느림 |
| 구현 복잡도 | - 비교적 단순 | - 파일 관리 및 병합으로 복잡 |

- 내부정렬은 메모리 내 연산 효율이 핵심이고, 외부정렬은 디스크 I/O 최소화가 성능의 핵심임

“끝”

| | | | |
|------|---|-----|-----------|
| 11 | 설계검리와 종료검리 | | |
| 문제 | 설계검리와 종료검리의 주요점검 사항 | | |
| 도메인 | 소프트웨어공학 | 난이도 | 중 (상/중/하) |
| 키워드 | 적정성 사전 점검, 구현 결과 검증 | | |
| 출제배경 | 전주기 품질관리 관점에서 사전 예방과 사후 검증 역할을 구분하고, 실무 감리 판단 능력 확인 | | |
| 참고문헌 | 서브노트, ITPE 기술사회 자료 참조 | | |
| 출제자 | 강평야 전일 기술사(제 114 회 정보관리기술사 / ji@naver.com) | | |

I. 설계 적정성과 결과 완성도를 검증, 설계 및 종료 검리의 개념

| | |
|--|---|
| <div style="display: flex; justify-content: space-around; align-items: center;"> <div style="text-align: center;"> <div style="background-color: #0056b3; color: white; padding: 5px; margin-bottom: 10px;">설계검리</div> <div style="background-color: #0056b3; color: white; padding: 5px; margin-bottom: 10px;">제대로 설계했는지</div>  <ul style="list-style-type: none"> • 요구사항 검토 • 설계 아키텍처 점검 • 오류 및 재작업 예방 <div style="border: 1px dashed #ccc; padding: 5px; margin-top: 10px;">설계 단계의 적정성 사전 점검</div> </div> <div style="font-size: 2em; color: #0056b3;">➔</div> <div style="text-align: center;"> <div style="background-color: #e67e22; color: white; padding: 5px; margin-bottom: 10px;">종료검리</div> <div style="background-color: #e67e22; color: white; padding: 5px; margin-bottom: 10px;">제대로 만들고 운영될 준비</div>  <ul style="list-style-type: none"> • 구현 결과 검증 • 테스트 및 결함 확인 • 운영 준비 점검 <div style="border: 1px dashed #ccc; padding: 5px; margin-top: 10px;">구축 완료 후 최종 확인</div> </div> </div> | |
| 설계검리 | - 요구사항을 기반으로 수립된 아키텍처 및 상세 설계가 타당하고 구현 가능한지를 사전에 점검하여, 개발 단계에서 발생할 수 있는 품질-리스크를 예방 하는 감리 |
| 종료검리 | - 시스템 구축 완료 후 구현 결과가 요구사항과 설계에 부합하는지를 검증 하고, 운영 가능성과 품질 완성도 를 최종 확인하는 감리 |

- 사업비나 기간이 일정 규모 미만일 경우 요구정의 단계 감리를 생략하고 2 단계 감리로 진행

II. 설계검리와 종료검리의 주요점검 사항

가. 설계검리 시 주요점검 사항

| 구분 | 주요 점검 항목 | 세부 점검 내용 |
|---------|------------|------------------------------------|
| 요구사항 반영 | 요구사항 충족 여부 | - 요구사항 정의서 대비 설계 반영 여부 |
| | 요구사항 추적성 | - 요구사항-설계 산출물 간 Traceability 확보 여부 |

| | | |
|----------|----------|-----------------------------|
| 아키텍처 설계 | 구조 적정성 | - 업무·데이터·애플리케이션·기술 아키텍처 정합성 |
| | 확장·성능 고려 | - 성능, 확장성, 가용성 설계 반영 여부 |
| 상세 설계 | 설계 완성도 | - 화면·프로그램·DB·인터페이스 설계 구체성 |
| | 예외 처리 | - 오류·예외·경계값 처리 설계 여부 |
| 비기능 요구사항 | 보안 설계 | - 접근통제, 권한관리, 개인정보 보호 반영 여부 |
| | 운영 안정성 | - 로그, 모니터링, 장애 대응 설계 여부 |
| 인터페이스 | 연계 정의 | - 연계 방식, 주기, 데이터 항목 정의 명확성 |
| | 오류 처리 | - 연계 오류 발생 시 처리·재처리 방안 |
| 테스트·품질 | 테스트 가능성 | - 설계 기반 테스트 케이스 도출 가능 여부 |
| | 결함 관리 | - 결함 관리 및 품질 개선 계획 반영 여부 |
| 표준·준수 | 표준 적용 | - 개발·설계 표준 및 법·지침 준수 여부 |

나. 종료감리 시 주요점검 사항

| 구분 | 주요 점검 항목 | 세부 점검 내용 |
|---------|------------|----------------------------|
| 요구사항 이행 | 요구사항 충족 여부 | - 요구사항 정의서 대비 구현 결과 충족 여부 |
| | 미이행 관리 | - 미반영·변경 요구사항에 대한 조치·승인 여부 |
| 시스템 구현 | 기능 구현 완성도 | - 설계서 대비 기능 구현 적정성 |
| | 오류·결함 | - 잔여 결함 및 조치 결과 적정성 |
| 테스트 수행 | 테스트 적정성 | - 단위·통합·시스템·인수 테스트 수행 여부 |
| | 결과 검증 | - 테스트 결과 및 증빙자료 적정성 |
| 비기능 품질 | 성능·보안 | - 성능시험, 보안 점검 결과 충족 여부 |
| | 안정성 | - 장애 대응, 복구, 백업 체계 검증 |
| 운영 준비 | 운영 이관 | - 운영 절차, 매뉴얼, 인수인계 적정성 |
| | 교육 수행 | - 사용자·운영자 교육 수행 여부 |
| 데이터 관리 | 데이터 정합성 | - 데이터 이행·정합성·무결성 검증 |
| 표준·준수 | 기준 충족 | - 법·지침·표준 및 계약 요건 충족 여부 |
| 산출물 | 산출물 완결성 | - 최종 산출물 제출 및 승인 여부 |

- 종료감리는 형식적 완료가 아닌, 실제 운영 가능성과 품질 완성도를 기준으로 판단해야 함

III. 설계감리와 종료감리의 비교

| 구분 | 설계감리 | 종료감리 |
|-------|-----------------------|--------------------------|
| 감리 목적 | - 설계 적정성·구현 가능성 사전 검증 | - 구축 결과 완성도·운영 가능성 최종 검증 |
| 감리 시점 | - 설계 단계(설계 종료 후) | - 구축 완료 후(운영 이행 전) |
| 중점 대상 | - 아키텍처·상세설계·비기능 설계 | - 구현 결과·테스트 결과·운영 준비 |
| 감리 성격 | - 품질 예방 중심 | - 품질 검증·확정 중심 |

- 설계감리는 구현 전 품질을 예방하는 감리이며, 종료감리는 구현 결과의 품질을 확정하는 감리

“끝”

| | | | |
|------|--|-----|----------|
| 12 | CRUD 매트릭스 | | |
| 문제 | CRUD 매트릭스의 구성요소와 작성 절차 | | |
| 도메인 | 데이터베이스 | 난이도 | 중(상/중/하) |
| 키워드 | 생성(Create), 이용(Read), 수정(Update), 삭제>Delete), 데이터모델링 | | |
| 출제배경 | CRUD 매트릭스 이해도 확인 | | |
| 참고문헌 | ITPE 기술사회 자료 | | |
| 해설자 | 강평야 전일 기술사(제 114 회 정보관리기술사 / ji@naver.com) | | |

I. 프로세스와 데이터간의 상관관계 표현을 위한 Matrix, CRUD Matrix 의 개요

가. CRUD Matrix 의 정의

- 정보시스템에서 수행되는 업무 기능과 데이터 객체 간의 생성·조회·수정·삭제 관계를 행렬 형태로 표현하여, 데이터의 생명주기와 업무 기능 간 **정합성·중복·누락 여부를 검증**하고 **일관성을 확보**하기 위한 분석 도구

나. CRUD Matrix 의 목적

| 목적 | 설명 |
|-------------|--|
| 모델링 작업 검증 | - 분석 단계의 데이터 모델과 프로세스 모델에 대한 작업을 검증하는 역할 |
| 테스트에서 사용 | - 테스트 단계에서 개발한 애플리케이션 테스트를 위한 객관적인 자료 |
| 인터페이스 현황 파악 | - 전체 업무의 인터페이스를 파악 |

- 업무-데이터 간 관계를 명확히 하여 정합성·누락·중복을 검증하기 위함

II. CRUD Matrix 구성요소와 작성 절차

가. CRUD Matrix 구성요소

| 구분 | 구성요소 | 설명 |
|-----------|-----------------------|-------------------------------|
| 행(Row) | 업무 기능 / 프로세스 | - 시스템에서 수행되는 기능, 유스케이스, 업무 단위 |
| 열(Column) | 데이터 객체(Entity/Object) | - 관리 대상 데이터(엔터티, 테이블, 객체) |
| 연산 | C(Create) | - 데이터 생성 |
| | R(Read) | - 데이터 조회 |
| | U(Update) | - 데이터 수정 |
| | D>Delete) | - 데이터 삭제 |
| 표기 규칙 | CRUD 기호 | - C/R/U/D 또는 CRU 등 복합 표기 |
| 참조(선택) | 프로그램·화면 정보 | - 화면 ID, 프로그램 ID, 트랜잭션 정보 등 |

- 각 구성요소는 업무 처리 흐름과 데이터 생명주기를 연결하는 기준 역할

나. CRUD Matrix 작성 절차

| 작성 절차 | 주요 활동 | 산출물 |
|--------------|-----------------------------|--------------|
| 업무 기능 도출 | - 업무분해를 통해 기능·프로세스 식별 | - 기능목록서 |
| 데이터 객체 정의 | - 데이터 모델에서 엔티티(객체) 도출 | - 데이터 모델 |
| 기능-데이터 관계 분석 | - 기능별 데이터 접근 유형(C/R/U/D) 분석 | - CRUD 관계 정의 |
| CRUD 매트릭스 작성 | - 기능×데이터 교차 영역에 CRUD 기입 | - CRUD 매트릭스 |

- . CRUD Matrix 작성 절차는 업무 기능과 데이터 객체를 식별하고, 각 기능이 데이터에 대해 수행하는 생성·조회·수정·삭제 관계를 분석하여 매트릭스로 정의하는 과정

III. CRUD Matrix 분석 사례

| 구분 | 분석방법(체크포인트) |
|--------------|--------------------------------------|
| 데이터모델 검증 | - 모든 엔티티에 대해 CRUD가 한번 이상 표시 되었는지 검증 |
| | - 모든 엔티티에 대해 한번이상 C가 표시 되었는지 검증 |
| | - 모든 엔티티에 대해 R이 한번 이상 존재하는지 검증 |
| 프로세스모델 검증 | - 모든 단위 프로세스는 하나 이상의 엔티티와 관련이 있는지 검증 |
| | - 두 개 이상의 엔티티가 동일한 엔티티를 생성하지는 않는지 검증 |

- 위 다섯 가지 사항이 모두 적절하다면 도출된 데이터 모델과 프로세스 모델은 적절한 관계에 있다고 할 수 있음

"끝"

| | | | |
|-------------|--|------------|-----------------|
| 13 | 성과경제(Outcome Economy) | | |
| 문제 | 성과경제(Outcome Economy)의 개념과 전통적 경제와의 차이점 | | |
| 도메인 | 경영전략 | 난이도 | 중(상/중/하) |
| 키워드 | 성과(Outcome) 중심 가치, 성과 기반 보상(Pay-for-Outcome), 디지털 기술 기반 측정 | | |
| 출제배경 | 전통적 공급 중심 경제의 한계를 넘기 위한 최신 경영 패러다임 이해 확인 | | |
| 참고문헌 | ITPE 서브노트, ITPE 모의고사 | | |
| 출제자 | 강평야 전일 기술사(제 114 회 정보관리기술사 / ji@naver.com) | | |

I. 결과 중심 가치 제공, 성과경제의 개념

가. 성과경제(Outcome Economy)의 정의

- 기업이 제품이나 서비스를 판매하는 데서 그치지 않고, **고객이 실제로 달성한 성과(Outcome)**에 대해 가치를 제공하고 그 결과에 따라 **보상**을 받는 경제 모델

나. 성과경제 등장배경

| 등장배경 | 설명 |
|---------------------|---|
| 제품·서비스 중심 가치의 한계 | - 기술·제품의 동질화로 차별화가 어려워지고, 단순 판매 중심 모델의 가치 창출에 한계 발생 |
| 고객의 성과·ROI 중심 요구 확대 | - 고객이 소유보다 실제 효과와 성과를 중시하며 결과 책임을 공급자에게 요구 |
| 디지털 기술 발전 | - IoT·데이터·AI 확산으로 성과 측정·모니터링이 가능해져 성과 기반 거래 실현 가능 |

- 이러한 배경 속에서 기업은 제품이나 서비스 제공에 그치지 않고, 고객이 실제로 달성한 성과에 대해 책임지고 가치를 제공하는 방식으로 비즈니스 모델을 전환

II. 성과경제와 전통적 경제와의 차이점

가. 개념 및 가치 관점의 차이

| 구분 | 전통적 경제 | 성과경제 |
|---------|------------------|--------------------|
| 경제 패러다임 | - 공급 중심 경제 | - 성과 중심 경제 |
| 가치의 기준 | - 무엇을 제공·판매했는가 | - 어떤 성과를 달성했는가 |
| 핵심 질문 | - 제품/서비스를 납품했는가? | - 고객의 목표가 달성되었는가? |
| 가치 인식 | - 소유 자체에 가치 부여 | - 사용 결과와 효과에 가치 부여 |
| 성과 측정 | - 제한적·비정형 | - 정량적·지속적 측정 필수 |
| 데이터 역할 | - 부가적 요소 | - 성과 판단의 핵심 요소 |

나. 비즈니스 및 운영 관점의 차이

| 구분 | 전통적 경제 | 성과경제 |
|--------|---------------|-----------------------------|
| 수익 구조 | - 일회성 판매·납품 | - 성과 기반 보상(Pay-for-Outcome) |
| 책임 범위 | - 계약·납품 시점까지 | - 성과 달성 시점까지 확대 |
| 계약 방식 | - 가격·수량 중심 계약 | - 성과 지표(SLA/KPI) 기반 계약 |
| 고객 관계 | - 거래 종료형 | - 지속적 파트너십 |
| 위험 부담 | - 주로 고객 부담 | - 공급자·고객 공동 부담 |
| 서비스 형태 | - 제품/서비스 제공 | - 성과 달성 지원 서비스 |

- 전통적 경제가 제품이나 서비스의 제공 자체를 중심으로 가치와 책임을 정의한다면, 성과경제는 고객이 실제로 달성한 성과를 기준으로 가치 창출과 보상, 책임 범위를 재정의하는 경제 모델

III. 성과경제 모델 사례

| 분야 | 성과경제 모델 사례 | 성과 기준 |
|----------|-------------------------------------|-----------------|
| 제조·항공 | - 엔진 가동시간 기반 서비스(Power by the Hour) | - 가동률, 장애 감소 |
| IT·디지털 | - 성과 기반 IT 서비스 계약 | - 업무 효율, 비용 절감 |
| 공공·스마트시티 | - Outcome-based 공공 계약 | - 혼잡 감소, 에너지 절감 |

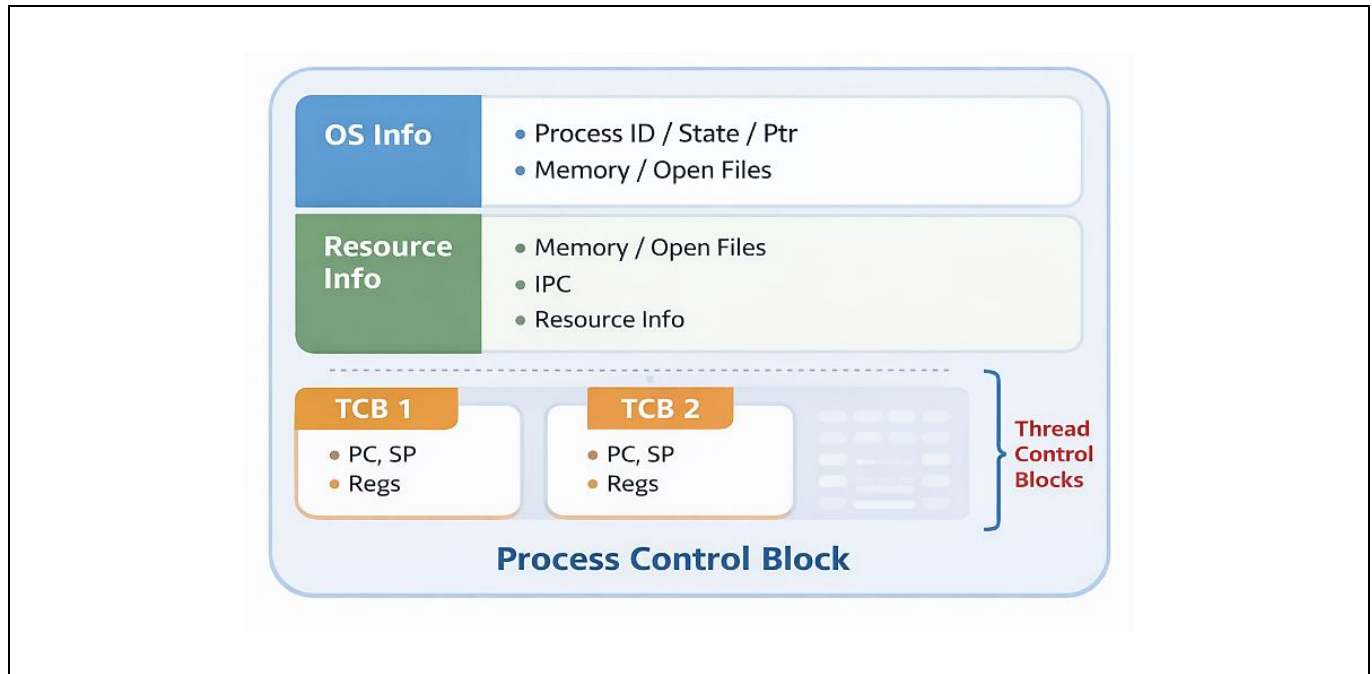
- 이러한 사례들은 제품이나 시스템을 제공하는 데서 그치지 않고, 고객과 사회가 실제로 달성한 성과를 기준으로 가치와 보상이 결정되는 성과경제 모델의 특징을 잘 보여줌

“끝”

| | | | |
|------|--|-----|----------|
| 13 | PCB(Process Control Block) 와 TCB(Thread Control Block) | | |
| 문제 | 운영체제에서 PCB(Process Control Block) 및 TCB(Thread Control Block)가 스케줄링에 활용되는 방식 | | |
| 도메인 | CA/OS | 난이도 | 중(상/중/하) |
| 키워드 | 상태, 문맥교환, 우선순위, 스케줄링 큐, CPU 사용량, 타임슬라이스, 응답시간 | | |
| 출제배경 | 멀티스레드 및 병렬 처리 환경에서의 운영체제 핵심 메커니즘 이해 확인 | | |
| 참고문헌 | ITPE 서브노트, ITPE 모의고사 | | |
| 출제자 | 강평야 전일 기술사(제 114 회 정보관리기술사 / ji@naver.com) | | |

I. Process Control Block 과 Thread Control Block 의 개요

가. PCB 와 TCB 의 연계 구조



- 스레드는 프로세스에 종속된 실행 단위이며, 하나의 프로세스는 하나 이상의 커널 스레드로 구성
- PCB 는 하나 이상의 TCB 를 포함하고 있으며, 프로세스가 종료되면 해당 프로세스에 포함된 모든 스레드도 함께 종료

나. PCB 와 TCB 의 정의

| 구분 | 정의 |
|---------------------------------------|--|
| PCB (Process Control Block) | - 운영체제가 프로세스의 실행 상태와 자원 정보를 관리하여 스케줄링과 문맥교환의 기준으로 사용하는 제어 블록 |
| TCB (Thread Control Block) | - 운영체제가 스레드의 실행 상태와 CPU 수행 정보를 관리하여 스레드 스케줄링과 경량 문맥교환의 기준으로 사용하는 제어 블록 |

- TCB 는 PCB 가 속한 프로세서 상태와 포인터 정보를 포함

II. 운영체제에서 PCB 및 TCB 가 스케줄링에 활용되는 방식

가. 운영체제에서 PCB 가 스케줄링에 활용되는 방식

| 구분 | 핵심 항목 | 스케줄링 관점 |
|---------|-----------------------------------|--------------------------|
| 관리 대상 | - 프로세스 | - CPU 를 간접적으로 할당 |
| 상태 정보 | - New / Ready / Running / Waiting | - 상태 기반 스케줄링 판단 |
| 실행 정보 | - PC, 레지스터, 주소 공간 | - Context Switch 시 전체 교체 |
| 스케줄링 정보 | - 우선순위, CPU 사용 시간 | - 스케줄링 정책 판단 근거 |
| 역할 요약 | - 자원 소유 및 관리 | - 스케줄링 기준 제공 |

나. 운영체제에서 TCB 가 스케줄링에 활용되는 방식

| 구분 | 핵심 항목 | 스케줄링 관점 |
|---------|-----------------------------|---------------------|
| 관리 대상 | - 스레드 | - CPU 직접 할당 대상 |
| 상태 정보 | - Ready / Running / Waiting | - 스레드 단위 스케줄링 |
| 실행 정보 | - PC, SP, 레지스터 | - 경량 Context Switch |
| 스케줄링 정보 | - 우선순위, 타임 슬라이스 | - 스케줄러 선택 기준 |
| 역할 요약 | - 실행 흐름 관리 | - CPU 스케줄링 실체 |

- PCB 는 프로세스 자원 관리를 위한 제어 블록이며, 실제 CPU 스케줄링은 TCB 단위로 수행되어 Context Switch 비용을 최소화함

III. PCB 와 TCB 비교

| 구분 | PCB | TCB |
|----------|------------------|-------------------|
| 스케줄링 실체 | - 논리적 스케줄링 기준 단위 | - 실제 CPU 실행 단위 |
| 주소 공간 관계 | - 독립 주소 공간 보유 | - PCB 의 주소 공간 공유 |
| 병렬성 관점 | - 프로세스 간 병렬 실행 | - 동일 프로세스 내 병렬 실행 |
| 스케줄링 효율 | - 전환 비용 큼 | - 전환 비용 작아 고효율 |

- PCB 는 스케줄링 판단의 기준 정보이며, TCB 는 CPU 가 직접 선택하는 실행 주체임

“끝”



제138회 대비 ITPE Final Round 해설집 (3일차)

| | |
|-----|---|
| 대 상 | 정보관리기술사, 컴퓨터시스템응용기술사, 정보통신기술사, 정보시스템감리사 시험 |
| 발행일 | 2026년 02월 02일 |
| 집 필 | 강정배 PE, 서O욱 PE, 조재원 PE, 이상헌 PE, 이제이 PE, 전일 PE, 김찬일 PE, 강진우 PE 장O호 PE, 백현 PE, 정상 PE |
| 출 판 | ITPE(Information Technology Professional Engineer) |
| 주 소 | 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.