ICT의 가치를 이끄는 사람들!!

126회

# 정보관리기술사 기출풀이 1교시

T 02.724.1831 E kpcitpe@kpc.or.kr H www.kpc.or.kr C cafe.naver.com/kpcitpe

# 국가기술자격 기술사 시험문제

정보처리기술사 제 126 회 제 1 교시

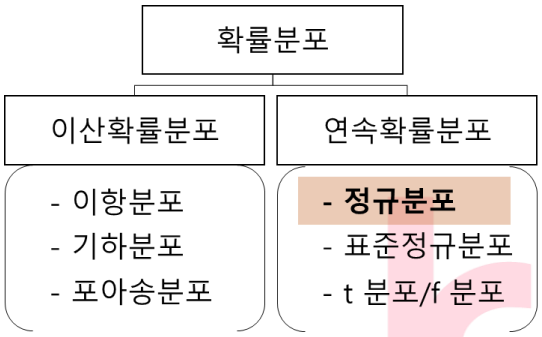
| 분야 | 정보처리 | 종목 | 정보관리기술사 | 수험번호 | | 성명 | |
|---|---|---|---|---|---|---|---|

※ 다음 문제 중 10 문제를 선택하여 설명하시오. (각 10 점)

1. 정규분포 특징
2. 메타휴리스틱스(Metaheuristics)
3. Race Condition
4. SNN (Spiking Neural Network)
5. 빅데이터 분석도구를 선택하는 원칙
6. 소프트웨어 품질인증
7. CAP 이론의 한계와 PACELC(Partition Availability Consistency Else Latency Consistency) 이론
8. 의사결정나무의 지니 지수(Gini Index)와 엔트로피 지수(Entropy Index)
9. 임베디드 소프트웨어 테스트(Embedded Software Test)
10. 스레싱(Thrashing)
11. 빅 엔디언(Big Endian) 과 리틀 엔디언(Little Endian)
12. 네트워크 스캐닝(Network Scanning)
13. 파일슬랙(File Slack)

| 문 제 | 1. 정규분포의 특징 | | |
|---|---|---|---|
| 출 제 영 역 | 알고리즘 | 난 이 도 | ★★☆☆☆ |
| 출 제 배 경 | - 데이터 분석을 위해 사용되는 통계 기본 개념으로 확률분포와 정규분포의 이해를 확인 | | |
| 출 제 빈 도 | 미출제 | | |
| 참 고 자 료 | - https://digital-play.tistory.com/75<br>- https://www.mathfactory.net/11300 | | |
| K e y w o r d | -연속확률분포, 평균, 분산, 표준정규분포, 중심극한정리(Central limit theorem, CLT) | | |
| 풀 이 | 김유리(124 회 정보관리기술사) | | |

## 1. 대표적인 연속 확률분포, 정규분포(Normal Distribution)의 개요



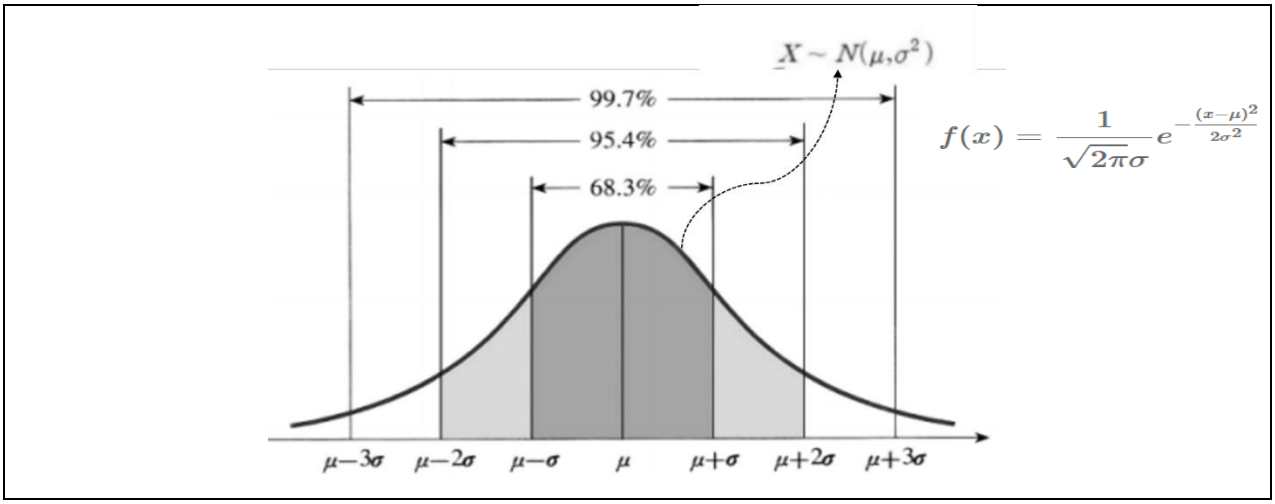

**정의**

- 평균 μ 와 분산 σ2 두 매개변수에 따라 구체적인 분포의 위치와 모양을 결정하는 좌우 대칭 종 모양 형태의 확률 분포

- 연속확률변수 X 의 확률밀도함수 f(x)가 $f(x)=\frac{1}{\sqrt{2\pi}\sigma}e^{-\frac{(x-\mu)^2}{2\sigma^2}}, -\infty<x<\infty$ 일 때, 이 확률분포를 정규분포라 하고, N(μ, σ2)으로 나타냄

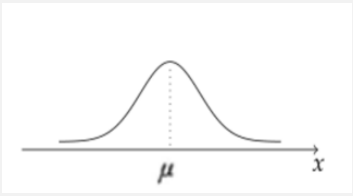
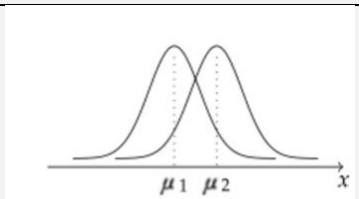
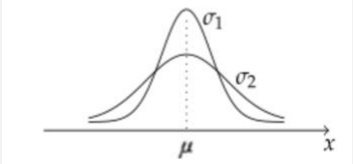
## 2. 정규분포(normal distribution)의 확률함수 및 특징

### 가. 정규분포(normal distribution)의 확률함수



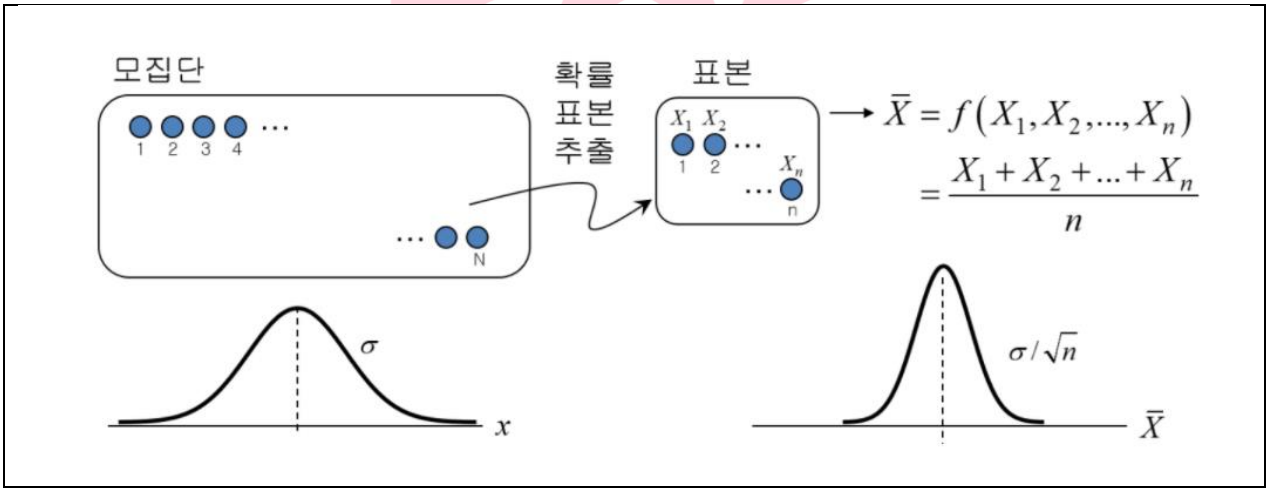

- μ 의 값은 분포의 중앙을 결정하고 σ 는 분포의 퍼짐 정도를 나타내며 μ ( - ∞ <μ< ∞ )와 σ ( ≥ 0 ) 의 값이 변함에 따라 다른 정규분포로 표현

### 나. 정규분포의 특징 설명

| 구분 | 개념도 | 특징 |
|---|---|---|
| 단순 |  | 1)평균을 기준으로 좌우대칭, 평균치에서 값이 가장 높은 최고점이며 최빈값, 중앙값, 평균이 모두 같음 |
| | | 2) x 축을 점근선으로 사용함 |
| | | 3)모든 곡선 아래의 전체 면적은 1 |
| | | 4)곡선은 평균으로부터 멀어질수록 x 축에 가까워지나, 결코 x 축에 닿지 않음 |
| 복합 |  | 5) σ이 일정할 때, μ의 값이 변하면 대칭축의 위치가 변함 (위 그림, μ이 변하는 경우 μ1<μ2) |
| |  | 6) μ의 값이 일정할 때, σ의 값이 커질수록 중앙 부분이 낮아지면서 양쪽으로 값이 작아질수록 곡선의 중앙부분이 높아지면서 좁아짐<br>(아래 그림, σ가 변하는 경우 σ1<σ2) |

- 모든 표본 평균의 분포가 표본의 크기가 커짐에 따라 정규분포에 유사한 형태로 변해간다는 정리 이론인 중심극한정리(Central limit theorem, CLT)에서 정규분포가 유용하게 사용됨

### 3. 중심극한정리(Central limit theorem, CLT) 에서의 정규분포의 사용




- 중심극한정리는 그 표본의 크기가 커질 수록 (보통 30 이상), 표본 평균들이 이루는 분포가 <모집단의 평균 μ 그리고 표준 편차가 σ/√n 인 정규분포> 에 가까워진다는 정리

"끝"

**기출풀이 의견**

1. 정규분포의 기본적인 개념을 묻는 문제로 확률분포 유형 제시, 개념, 특징을 작성해주시면 됩니다. 또한, 표준정규분포나 향후 출제가능한 중심극한정리(Central limit theorem, CLT)를 3단락으로 제시하시면 좋겠습니다.

| 문 제 | 2. 메타휴리스틱스(Metaheuristics) | | |
|---|---|---|---|
| 출 제 영 역 | 알고리즘 | 난 이 도 | ★★★★★ |
| 출 제 배 경 | -데이터 분석에서 각종 AI 알고리즘 이용이 높으며 그중 유전 알고리즘처럼 성능면에서 효율적인 분석 기법에 대한 전반적 개념 이해 숙지 필요. | | |
| 출 제 빈 도 | 미출제 | | |
| 참 고 자 료 | - http://dmqm.korea.ac.kr/activity/seminar/228<br>- https://fintecuriosity-11.tistory.com/359?category=840181 | | |
| Keyword | -최적해, 유전자알고리즘 | | |
| 풀 이 | 김유리(124 회 정보관리기술사) | | |

## 1. 탐색 프로세스의 방향을 제시해주기 위한 전략, 메타휴리스틱스(Metaheuristics)의 개요

| 정의 | 특징 |
|---|---|
| - 특정 문제에 한정되지 않고 어떠한 문제에 대해서도 범용적으로 대응할 수 있도록 설계된 알고리즘의 기본적인 프레임워크 | 1) 좋은 해 탐색과 지역 최적에서 벗어나 탐색할 수 있는 프로세스, 전략을 갖는 해법<br>2) '자연 현상'을 모방하며, 과거 정보를 '기억'하거나 '전달'하여 다음 탐색에 이용<br>3) '단일해 기반(single-solution based)'혹은 '해 집단 기반(population-based)'으로 해 탐색 과정을 매 반복함 |

- 특정 문제의 해결을 위한 알고리즘을 설계할 때, 모든 메타휴리스틱스에서 공통적으로 해의 표현, 목적함수, 초기해, 파라미터 조정, 종료조건 등이 고려되는 기본 요소임

## 2. 메타휴리스틱스(Metaheuristics) 알고리즘의 설명

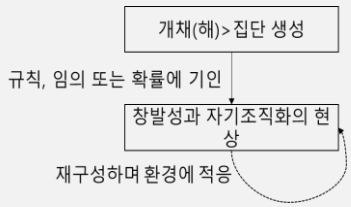
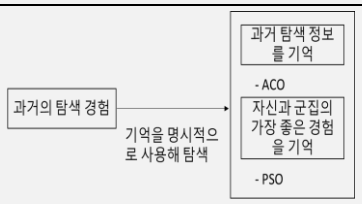
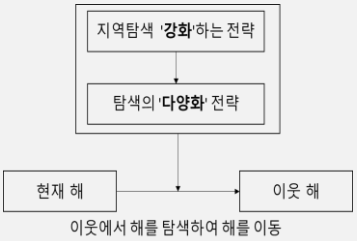
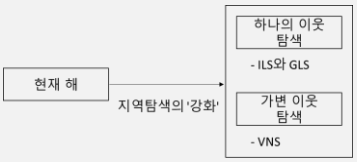
### 가. 메타휴리스틱스(Metaheuristics) 알고리즘의 공통요소 설명

| 공통요소 | 설명 |
|---|---|
| 표현방법 | - 이진수, 실수 벡터, 이산 값 벡터, 순열, 그룹, 행렬, 랜덤 키(Random key), 그래프 등 다양한 표현이 가능 |
| 목적함수 | - 후보해들의 품질, 적합도가 결정되는 목적함수/평가함수가 필요<br>- 목적함수는 좋은 해가 있는 탐색 영역으로 탐색을 유도하는 역할<br>- 좋은 해로의 탐색을 촉진하기 위하여 목적함수를 변형한 '평가함수'를 사용함 |
| 초기해 | - 생성 방법은 임의(random) 생성과 휴리스틱(또는 greedy)생성으로의 초기해 사용 |
| 파라미터 조정 | - 알고리즘의 유연성과 강건성을 재고시키는 역할을 하는 반면에, 적절하지 않은 파라미터 값은 알고리즘의 성능을 크게 약화<br>- 실행하기 전에 실험을 통해 파라미터 값을 고정시키는 방법<br>- 실행 중에 파라미터 값을 동적으로 또는 탐색에 적응하며 갱신하는 방법 |

| 종료조건 | - 알고리즘이 무한 반복을 하지 않고 합리적인 반복 횟수를 수행한 후 끝낼 수 있도록 종료 조건을 명시 |
|---|---|

- 메타휴리스틱 알고리즘은 동물의 진화론에서 영감을 얻어 1970년대에 개발된 유전자 알고리즘을 시작으로, 현재까지 자연 생태계 등에서 영감을 얻어 만들어진 다양한 방법론이 등장

**나. 메타휴리스틱스(Metaheuristics) 알고리즘의 설명**

| 구분 | 개념도 | 상세 유형 |
|---|---|---|
| 자연의 진화 과정을 모방 |  | - 진화전략(evolution strategy: ES)<br>- 진화프로그래밍(evolutionary programming: EP)<br>- 유전알고리즘(genetic algorithm: GA)<br>- 유전프로그래밍(genetic programming: GP)<br>- 차분진화(differential evolution: DE) |
| 생물들의 행동을 모방 |  | - 개미군체최적화(ant colony optimization: ACO)<br>- 입자군집최적화(particle swarm optimization: PSO)<br>- 벌군체최적화(Bee Colony Optimization: BCO)<br>- 인공벌군체(artificial bee colony: ABC) |
| 열역학의 어닐링 과정을 모방 |  | - 시뮬레이티드 어닐링(simulated annealing: SA)<br>- 기억 과정을 모방한 타부서치(tabu search: TS)<br>-면역시스템을 모방한 인공면역시스템(artificial immune system: AIS)<br>- 음악의 화음 조율과정을 모방한 화음탐색(harmony search: HS) 알고리즘 |
| 체계적인 반복으로 이웃을 탐색 |  | -복지역탐색(iterated local search: ILS)<br>-가변이웃탐색(variable neighborhood search: VNS)<br>-유도지역탐색(guided local search: GLS) |

- 대규모 데이터 셋과 탐색영역이 주어질 경우 메타휴리스틱만의 성능을 보장할 수 없기 때문에 병렬연산 처리능력 강화된 GPU, APU, 고성능 컴퓨터와 그에 따른 성능 개선 위한 메타휴리스틱 내재적인 알고리즘 변경도 필요함

### 3. 메타휴리스틱스 병렬화 방법에서의 고려사항

1) 매 탐색 시 주어진 임무(task)를 독립적인 부분임무(subtask)로 나눌 수 있는지,
2) 부분집합 합 문제(subset sum problem)에 해당하는지,
3) 병렬처리 효과를 반감시키는 특성(예시. 잦은 동기화 또는 과도한 중복 탐색)은 없는지,
4) 병렬처리가 탐험(exploration)과 활용(exploitation) 능력을 개선시키는지,
5) 병렬처리를 통해 알고리즘의 속도(speed)와 해의 질(quality of solutions)의 개선이 있는지

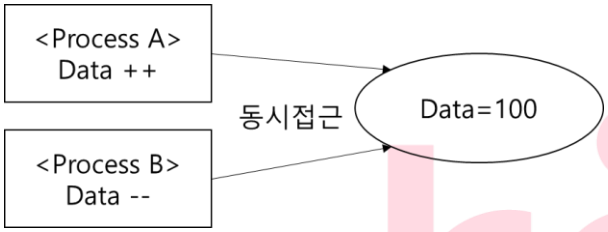
- 메타휴리스틱 알고리즘은 산업영역(일정관리, SCM, 공정 최적화), 컴퓨터과학영역(Internet Routing, Web page Clustering), 헬스케어 분야(RNA 구조 예측) 등 다양한 영역에서 활용 중

## 기출풀이 의견

2. 메타휴리스틱에 대한 개념과 유전알고리즘 포함의 유형 제시면 충분히 고득점 답안이 될 수 있습니다.

| 문 제 | 3. Race Condition | | |
|---|---|---|---|
| 출 제 영 역 | CA/OS | 난 이 도 | ★★★☆☆ |
| 출 제 배 경 | - Race Condition 의 원리가 공격에 이용되고 있어 개념, 요구조건 및 해결 방안 대한 이해 필요 | | |
| 출 제 빈 도 | 미출제 | | |
| 참 고 자 료 | - https://m.blog.naver.com/hirit808/221762802800<br>- https://velog.io/@woounnan/SYSTEM-Race-Condition-Attack | | |
| K e y w o r d | -상호배제, 뮤택스, 세마포어, IPC, 스핀락 | | |
| 풀 이 | 김유리(124 회 정보관리기술사) | | |

## 1. 병행프로세스 동기화 문제, 경쟁 상태(Race Condition)의 개요



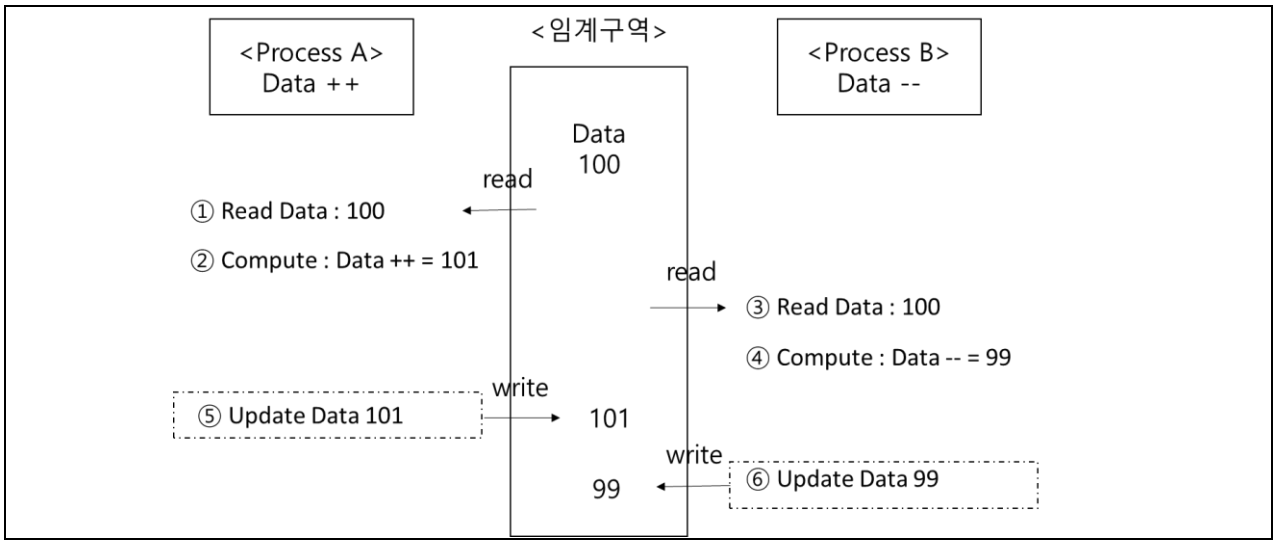

**정의**

- 여러 개의 프로세스들이 임계영역 내 공유 자원에 동시적(concurrent)으로 접근할 때, 접근 타이밍과 순서에 따라 그 결과가 달라져서 데이터 일관성(Data Consistency)을 해칠 수 있는 상태

- 멀티 프로세스 환경에서 공통 자원을 병행하여 작업할 때 Mutual Exclusion, Deadlock, Starvation 문제에 직면하며 해결을 위해 조건을 만족하는 기법을 이용해야 함

## 2. 경쟁 상태(Race Condition)의 발생 프로세스 및 요구조건과 해결방안

### 가. 경쟁 상태(Race Condition)의 발생 프로세스



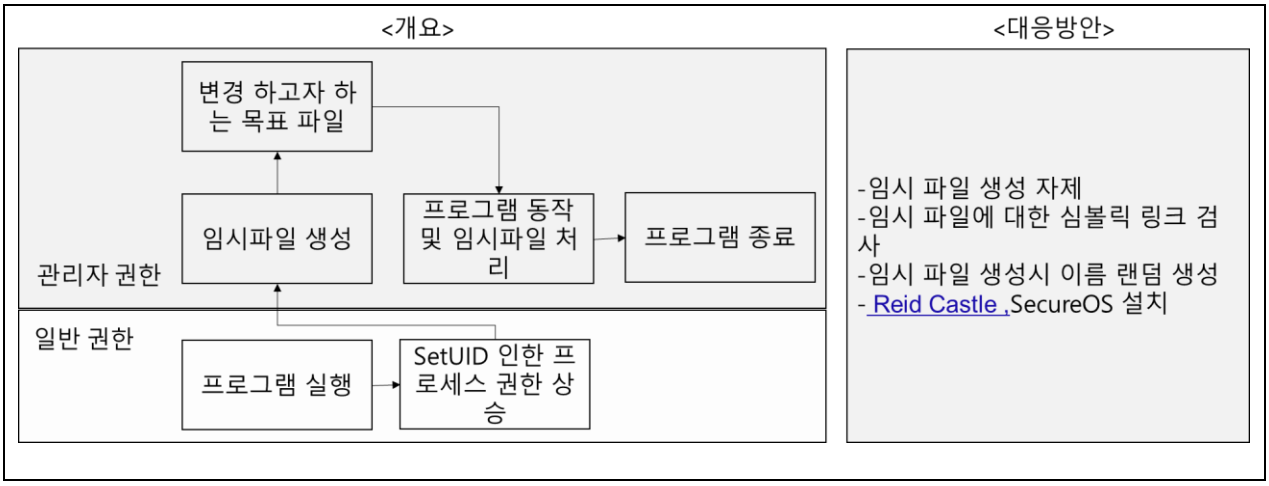

- 프로세스 A, B 가 동시에 공유자원 Data 를 사용할 때, 사용 순서에 따라 결과값이 달라지는 상태 발생
- Race Condition 해결을 위해 상호배제 등의 조건을 만족하며 SW, HW 측면의 기법을 이용하여 프로세스 동기화 구현이 되어져야 함

나. 경쟁 상태(Race Condition)의 요구조건과 해결방안

| 구분 | 기술 | | 설명 |
|---|---|---|---|
| 요구조건 | 상호 배제(Mutual Exclusion) | | - 특정 프로세스가 공유자원을 사용하고 있을 경우 다른 프로세스의 해당 공유자원의 접근을 제한하는 기법 |
| | 진행(Progress) | | - 임계영역에 있는 프로세스 외에는 다른 프로세스가 임계 영역에 진입하는 것을 방해하면 안됨 |
| | 한계대기(Bounded Waiting) | | - 기아(starvation) 상태를 방지하기 위해 프로세스가 임계 영역에 들어가려고 요청한 후부터 다른 프로세스들이 임계 영역에 들어가는 횟수에 한계가 있어야 하며 임계 영역에 한 번 들어갔다 나온 프로세스는 다음에 들어갈 때 제한 줌 |
| 해결방안 | SW | 데커(Dekker) 알고리즘 | - flag(누가 임계영역에 진입할 것인지 알리는 변수), turn(누가 임계영역에 들어갈 차례인지 알리는 변수) 구성<br>- 프로세스 두 개 일 때 상호 배제를 보장하는 최초의 알고리즘 |
| | | 피터슨(Peterson) 알고리즘 | - 데커의 알고리즘과 유사하지만 상대방에게 진입 기회를 양보한다는 차이가 있고 보다 더 간단하게 구현 |
| | | 램포트(Lamport) 알고리즘 | - 프로세스 n개의 상호 배제 문제를 해결한 알고리즘<br>- 프로세스 고유번호로 진입 우선순위 할당 |
| | HW | Test and Set 명령 | - 임계영역 진입 시, Lock 잠금 : Test and Set(Lock)<br>- 종료 후 Lock 해제 : Lock=0 |
| | | SWAP 명령 | - HW 상 두 메모리 워드간 내용의 원자적 교환<br>- Boolean *a(전역변수), Boolean *b(지역변수)간 교환 |
| | 동기화 | 세마포어 | - P(S):Wait 연산, V(S):Signal 연산 기반의 동기화 기법<br>- 바이너리 세마포어, 카운트 세마포어 |
| | | 모니터 | - 공유자원과 제어 메소드(wait, signal)을 묶어 관리<br>- 특정목적의 공유자원+뮤텍스/세마포어 추상화 데이터타입 |
| | | 뮤택스 | - 공유자원에 대한 쓰레드의 접근제어<br>- Lock 과 Unlock 상태로 변수 변경하며 자원접근 허용 |
| | | 스핀락 | - 프로세스가 사용하려는 Lock 을 획득할 때까지 명령어 루프를 돌며 계속 시도하는 Locking 기법 |

- Race Condition 원리 기반의 Race Condition Attack 은 프로세스간 경쟁을 이용하여, 관리자 권한을 얻는 공격방법이 있어 임시 파일 생성을 자제하는 등의 대응책이 필요함

### 3. Race Condition Attack 의 개요와 대응방안




- 그 외로 임시파일이 저장되는 디렉토리의 umask 를 최하 022 로 유지하여 임시로 생성한 파일이 공격자에 의해 삭제되지 않도록 하는 방법으로도 해결 가능

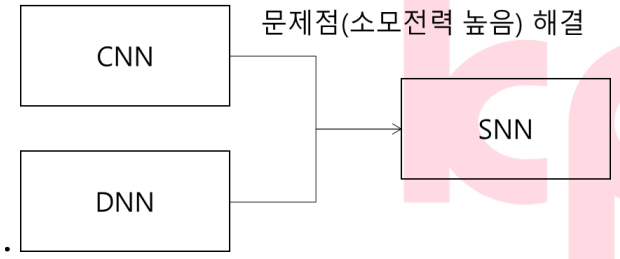
"끝"

**기출풀이 의견**

3. 기본적인 토픽으로 Race Condition 개념, 발생조건, 해결방안 제시 후 마지막으로 Race Condition Attack 공격 설명과 대응 방안을 제시해주시면 됩니다.

| 문 제 | 4. SNN (Spiking Neural Network) | | |
|---|---|---|---|
| 출 제 영 역 | 인공지능 | 난 이 도 | ★★★☆☆ |
| 출 제 배 경 | - 대용량의 데이터들이 쏟아지면서 현재 컴퓨팅 시스템의 한계를 직면, 해결을 위해 인간의 뇌를 모방기술한 신경망 기술에 대한 개념 확인 | | |
| 출 제 빈 도 | 미출제 | | |
| 참 고 자 료 | - https://www.itfind.or.kr/publication/regular/weeklytrend/weekly/view.do?boardParam1=8189&boardParam2=8189<br>- http://www.koreascience.kr/article/JAKO202025465017052.pdf<br>- http://www.tta.or.kr/data/androReport/ttaJnal/188-1-3-3.pdf | | |
| K e y w o r d | - 스파이크, 저전력 뉴로모픽 칩, STDP, 스파이크간 시간차이, 스파이크 출력 | | |
| 풀 이 | 김유리(124 회 정보관리기술사) | | |

## 1. 뉴로모픽 핵심 기술, SNN(Spiking Neural Network)의 개요



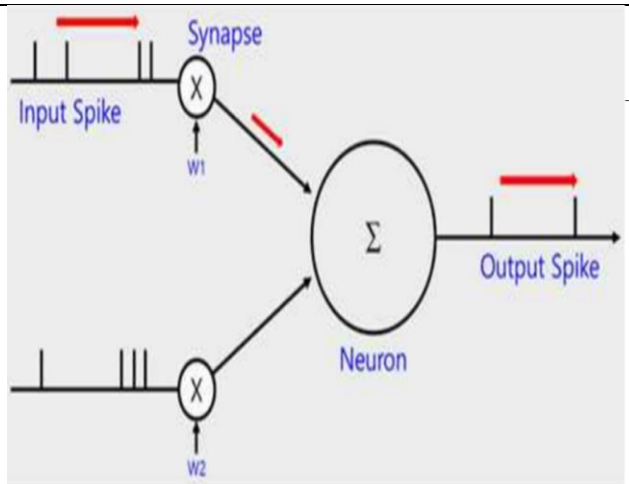

**정의**

- 두뇌에서 실제로 정보가 전달되고 가공되는 과정을 모사하여, 대량의 데이터를 상대적으로 적은 수의 스파이크로 처리할 수 있게 스파이킹 뉴런을 기반으로 설계된 신경망

- ① High performance, ② Low power, ③ On-chip learning 원칙 기반으로 두뇌의 생물학적 동작 구조를 모방 방식

## 2. SNN(Spiking Neural Network) 구현 모델과 학습방법

### 가. SNN(Spiking Neural Network) 구현 모델, LIF(Leaky Integrated-and-Fire) 의 개념도




LIF모델의 수식화

$$\tau\frac{dv}{dt} = (v_{rest} - v) + g_e(E_{exc} - v) + g_i(E_{inh} - v)$$

τ = 시간상수,
vrest= 휴지기의 막전위,
ge, gi= 각 흥분성, 억제성 시냅스의 전도도,
Eexc, Einh = 각각 평형상태의 흥분성, 억제성 시냅스의 전위

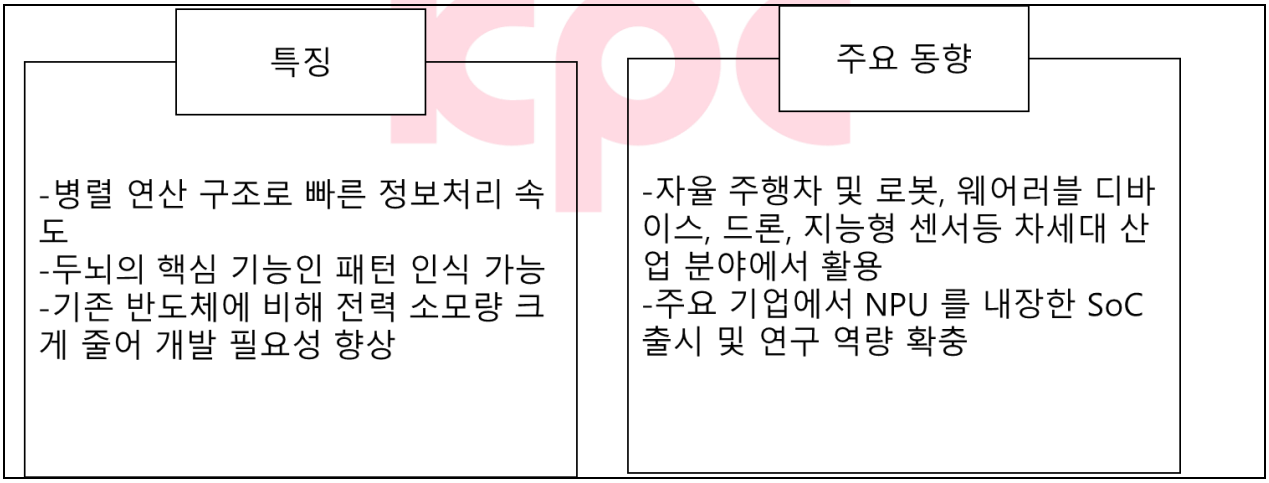
- 입력 스파이크는 각 시냅스의 가중치를 곱하여 뉴런에서 합쳐지게 되고, 합쳐진 막 전위 (membrane potential) 값이 문턱전압 (threshold voltage)를 넘게 되면 출력 스파이크를 다른 뉴런으로 보내고 뉴런내부의 막전위는 초기화하는 방식

**나. LIF(Leaky Integrated-and-Fire) 모델 처리 절차 및 학습 방법의 설명**

| 구분 | 개념도 | 설명 |
|---|---|---|
| **LIF 처리 절차** | 스파이크 입력 → 전위 증가 → 스파이크 발생 및 출력 | 1) 스파이크 입력<br>2) 시간적 간격과 형태에 따라 전위 증가<br>3) 일정수준 이상(문턱전압) 전위 도달 시 스파이크 발생 및 출력 |
| **학습 방법**<br>**- STDP (Spike-Timing-Dependent Plasticity)** | $\Delta w_{ij} = \sum_{pre}\sum_{post} W(w_{ij}, \Delta t)$ | - 시냅스를 사이에 두고 신호를 보내는 쪽인 시냅스 전 뉴런과 신호를 받는 쪽인 시냅스 후 뉴런 사이의 스파이크 발생 시간 상 관관계를 이용하여 시냅스 가중치를 조절 방식<br>- Δt 는 시냅스 후 뉴런과 시냅스 전 뉴런 사이의 스파이크 발생 시간차이로 작을수록 두 뉴런은 밀접한 관계를 표현 |

- HW 측면에서도 인간의 뇌 구조를 모방하여 구현하는 뉴로모픽 컴퓨팅칩에 대한 연구를 진행 중

### 3. 뉴로모픽 컴퓨팅칩의 특징 및 동향




- HW, SW 연구 결과가 융합되어 진보된 인공지능 구현에 활용된다면 현재 성능을 월등히 넘는 인공지능 구현의 가능성이 높아질 것으로 예상함

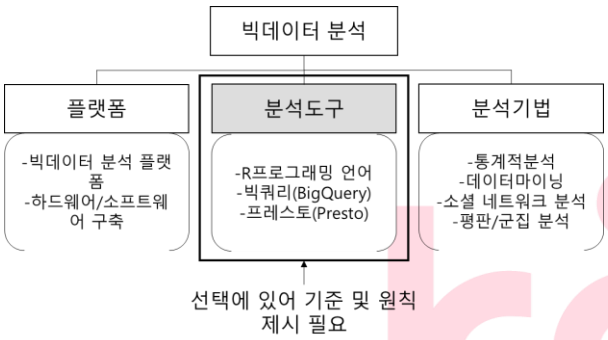
"끝"

**기출풀이 의견**

4. 기존 심층 신경망(DNN)의 성능 개선하고자 생물학적 신경 네트워크 구조를 유사하게 모방한 SNN 대한 등장 배경, 개념, 프로세스, 학습 모델등을 제시 해주시고 3단락으로 DNN의 비교나 뉴모로픽 칩의 동향과 연계한 답안을 작성해주시면 좋겠습니다.

| 문 제 | 5. 빅데이터 분석도구를 선택하는 원칙 | | |
|---|---|---|---|
| 출 제 영 역 | DB | 난 이 도 | ★★☆☆☆ |
| 출 제 배 경 | - 빅데이터 분석과 활용이 중요해지면서 분석도구를 선택함에 있어서 따라야할 원칙을 제시 | | |
| 출 제 빈 도 | 미출제 | | |
| 참 고 자 료 | - https://brunch.co.kr/@choobo/142<br>- https://finereport.com/kr/빅데이터-분석-프로그램/ | | |
| Keyword | - 빅데이터 분석, 빅데이터 활용, 소셜 네트워크 | | |
| 풀 이 | 김유리(124 회 정보관리기술사) | | |

## 1. 데이터를 효율적으로 분류, 예측 분석, 빅데이터 분석도구의 개요



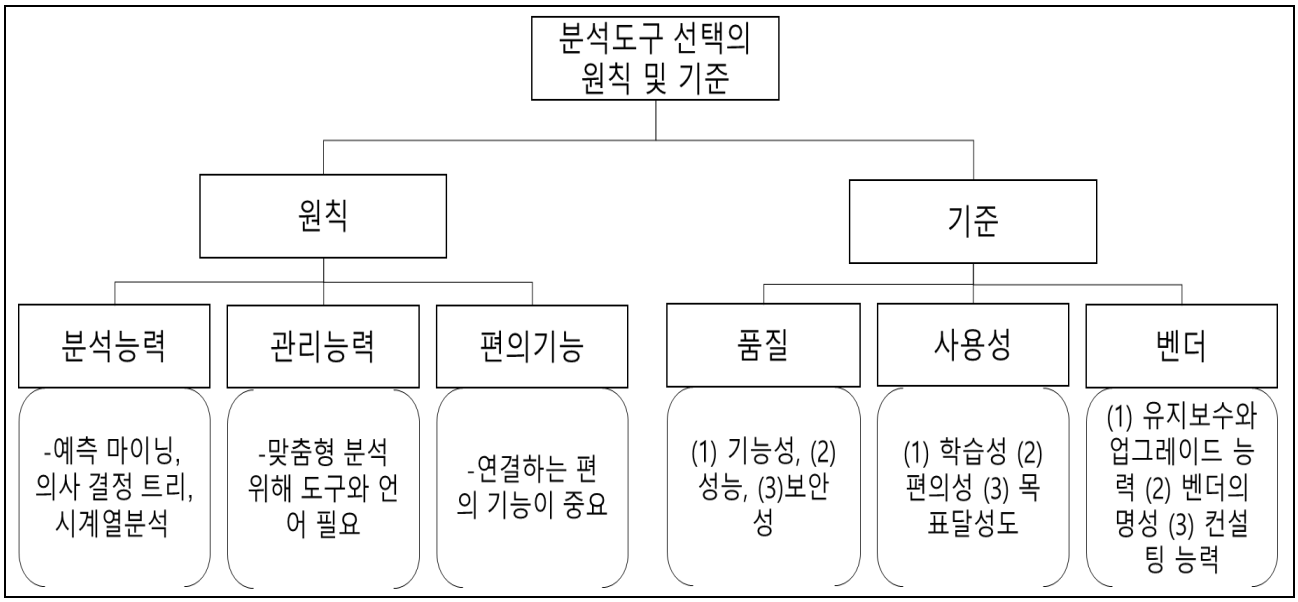

**정의**

- 고용량의 데이터를 예측기법, 데이터 마이닝, 통계학, 인공지능, 자연어 처리에 기반을 두고 빠른 속도로 분석하여 문제해결과 의사결정을 하는 중요한 도구

- 분석도구들의 특성을 파악하여 기업 및 다양한 기관에서 보유한 데이터들의 분석을 통해 데이터 가치를 높이고 활용성을 높이기 위해 선택하기 위한 원칙, 기준 마련 필요

## 2. 빅데이터 분석도구를 선택하는 원칙의 개념도 및 설명

### 가. 빅데이터 분석도구를 선택하는 원칙 및 기준의 개념도




- 빅데이터 도구를 도입하기 전에, 목적, 기능, 보안, 비용성능 등을 고려하여 선택해야 함

나. 빅데이터 분석도구를 선택하는 원칙 및 기준의 설명

| 구분 | 세부 원칙 및 세부 기준 | 설명 |
|---|---|---|
| 원칙 | 분석능력 | - 예측 마이닝, 의사 결정 트리, 시계열, 신경망, 경로 분석, 시장 바구니 분석 및 링크 분석을 포함한 다양한 유형의 분석을 위한 다양한 유형의 분석 기능제공 |
| | 관리능력 | - 다른 형태의 맞춤형 분석을 하기 위해서 조직이나 기업에서는 별도 통계 도구와 프로그래밍 언어(예: R)가 필요 |
| | 편의기능 | - 다양한 툴에서 데이터를 가져오거나 내보내는 것이 중요한 기능이며, 빅데이터 분석 도구를 빅 데이터 저장소에 연결하는 것이 얼마나 어려운(또는 쉬운)지를 이해하는 것이 핵심 고려 사항 |
| 기준 | 품질 | 1) 기능성, (2) 성능, (3)보안성 |
| | 사용성 | (1) 학습성 (2) 편의성 (3) 목표달성도 |
| | 벤더 | (1) 유지보수와 업그레이드 능력 (2) 벤더의 명성 (3) 컨설팅 능력 |

- 각각의 서비스 유형과 특성에 따라 다양한 형태로 분석결과를 제시위해 목적에 따른 분석 도구 선택 필요함

3. 목적에 따라 분석 도구의 유형

| 구분 | 분석 도구 | 상세 유형 |
|---|---|---|
| 검색 플랫폼에서 제공 | 구글의 구글 트렌드 | -글로벌 트랜드의 특정 검색어 비중 확인 유용 |
| | 네이버의 데이터랩 | -국내 트랜드의 특정 검색어 비중 확인 유용 |
| 전통적 MS 엑셀 제공 | 파워 쿼리 | -분석요구에 부합하게 데이터 검색, 연결, 결합등의 연결 기술 |
| | 파워 피벗 | -데이터 모델 생성, 관계 설정, 계산까지 가능한 데이터 모델링 기술, BI(비지니스 인텔리전스) 구현 가능 |
| 전문적 통계 패키지 | SPSS | -GUI 제공, 엑셀과 같은 메뉴 구조, 고급 통계 분석, 데이터 마이닝 가능 |
| | SAS | -프로그래밍 필요, 강력한 통계 분석 기능 제공 |
| 오픈 소스 프로그래밍 언어 | R | -통계분석 특화된 언어로 그래픽을 위한 언어, 벡터, 행렬, 테이블등 고유의 자료형 이용 |
| | 파이썬 | -개발에 특화된 언어, 시스템을 효과적으로 통합가능 |
| 오픈 소스 플랫폼 | 래피마이너(RapdMiner) | -GUI 방식으로의 데이터 마이닝 가능 분석도구 |
| | 나임(Knime) | -Work-flow 기반으로 분석 과정을 시각화 |

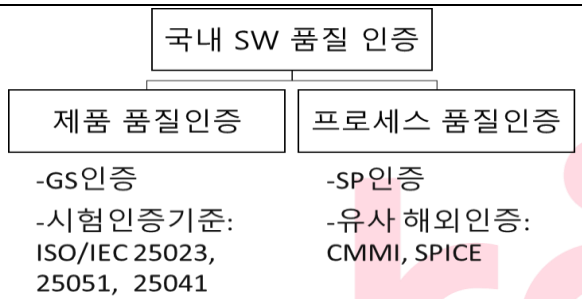
- 데이터 분석 결과를 해석하여 통찰을 얻는데 가장 효과적 방법은 데이터 시각화로 도식화 하는 기법이며, BI 위한 대시보드 역할을 하며 한눈에 동태적으로 데이터 분석의 결과를 제공가능 "끝"

**기출풀이 의견**

5. 빅데이터 분석과 활용이 중요해지면서 분석도구를 선택함에 있어서 따라야할 원칙을 제시하는 문제로 정해진 답을 몰라도 분석 도구를 선택함에 있어서 요구가 될 만한 내용으로 답안을 작성하셔도 되겠습니다.

| 문 제 | 6. 소프트웨어 품질인증 | | |
|---|---|---|---|
| 출 제 영 역 | 소프트웨어 공학 | 난 이 도 | ★★★☆☆ |
| 출 제 배 경 | - 4 차 산업혁명 시대 소프트웨어(SW) 시장 확대 함께 SW 품질에 대한 이슈가 증가하고 있으므로 품질인증의 필요성 이해 | | |
| 출 제 빈 도 | 123 회 컴퓨터시스템응용 | | |
| 참 고 자 료 | - SP 인증제도 소개 \| SWIT 소프트웨어산업정보시스템<br>-http://tta.or.kr/testlab/file/GS_intro.pdf | | |
| K e y w o r d | - GS 인증, SP 인증, ISO25023,25051,25041, CMMI, SPICE | | |
| 풀 이 | 김유리(124 회 정보관리기술사) | | |

## 1. SW 품질 향상, 소프트웨어 품질인증의 개요



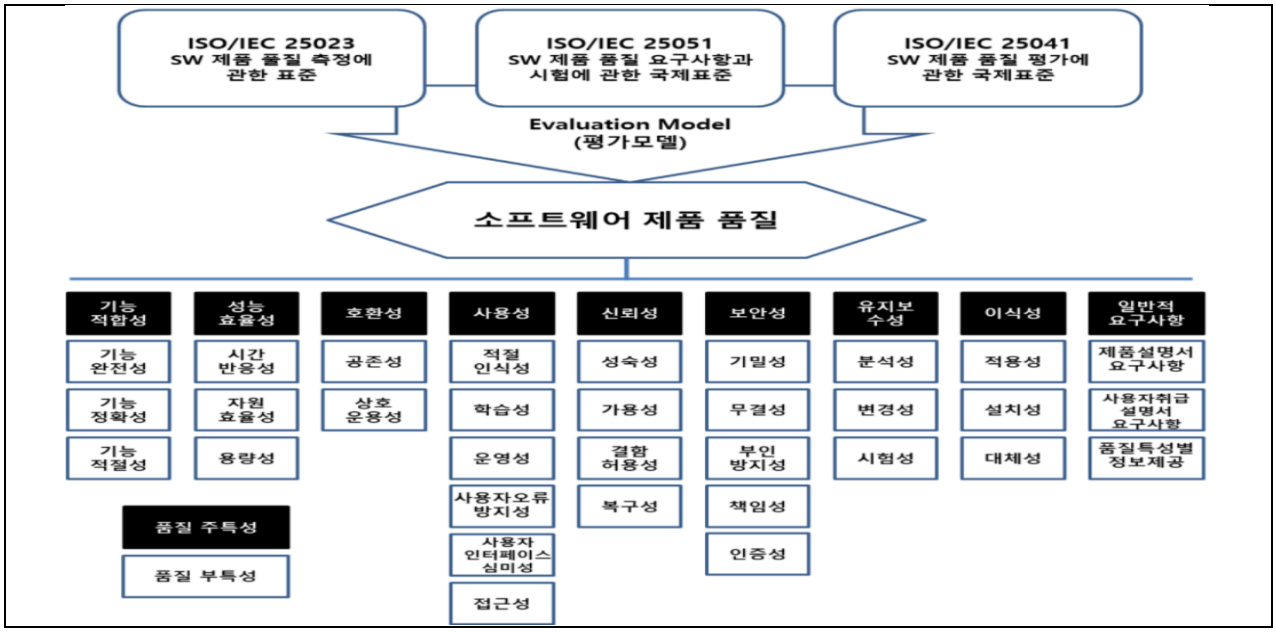

**정의**

- 소프트웨어의 품질 확보 및 유통 촉진을 목적으로 소프트웨어 제품 및 프로세스에 대한 품질특성을 국제표준 시험평가방법에 따라 평가 인증 활동.

- 프로젝트가 대형화/복잡화되고, 소프트웨어의 중요성이 높아짐에 따라 소프트웨어 품질의 중요성이 커짐
- 소프트웨어의 품질 인증은 소프트웨어의 제품 품질을 평가하여 인증하는 것과, 소프트웨어 개발 프로세스를 평가하여 인증하는 것으로 나눌 수 있음

## 2. 소프트웨어 품질인증의 분류 및 설명

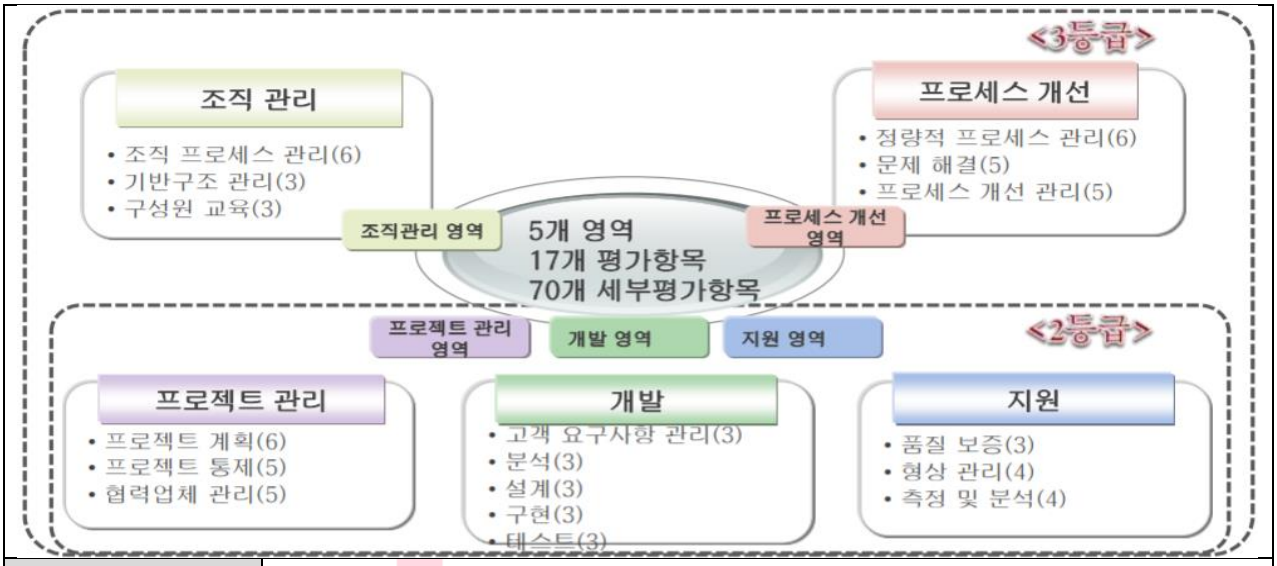
### 가. SW 제품 품질 인증, GS 인증의 설명



| 개념 | - 제품이 사용될 실제 운영환경의 테스트 시스템을 갖추어 소프트웨어에 대한 품질 특성을 국제 표준의 시험 평가방법에 따라 종합 평가하여, 설정 기준 이상의 품질 수준의 보유 여부를 확인하는 인증 제도, 소프트웨어 진흥법 제 20 조, |
|---|---|

- GS 인증은 소프트웨어 품질 시험으로, 다른 목적으로 수행되는 소프트웨어 보안 약점 진단, 정보보호 관련 인증(CC 인증) 등을 대체할 수 없음

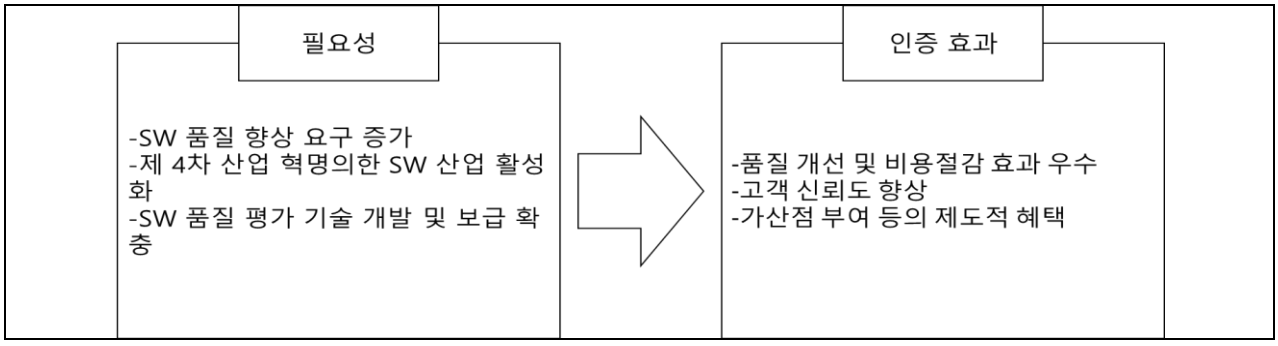
**나. SW 개발 공정의 품질 인증, SP 인증의 설명**




| 개념 | - 국내 SW 기업의 SW 사업 수행능력을 강화하고, 사업의 부실방지를 목적으로 기업의 SW 개발 단계별 작업 절차 및 산출물 관리 역량 등을 분석하여 SW 개발 프로세스 역량수준을 평가 인증하는 제도, 소프트웨어 진흥법 제 21 조 |
|---|---|

- 조직의 프로세스 역량 수준을 개선하기 위한 해외 인증 모델 CMMI, SPICE 와 달리 SP 인증은 조직의 프로세스 역량 수준을 평가하기 위한 모델임
- SP 인증은 소프트웨어 프로세스에 대한 품질을 인증할 뿐, 소프트웨어의 보안성까지 인증하지는 않음

### 3. SW 품질 인증 제도의 필요성 및 효과



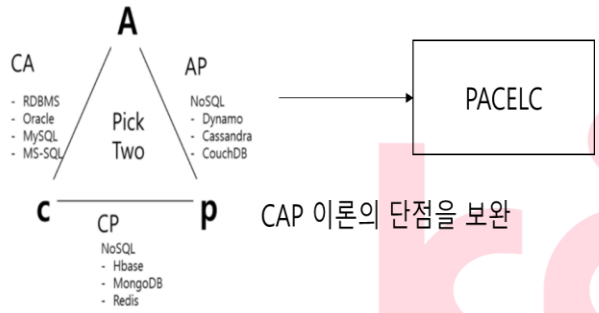

- SW 품질 인증 제도는 제품의 신뢰성 제고 및 국제경쟁력 확보에 이바지하고 있음. "끝"

**기출풀이 의견**

6. 기본적인 토픽으로 4차 산업혁명 시대 소프트웨어(SW) 시장 확대 함께 SW 품질에 대한 이슈가 증가하고 있으므로 품질인증의 필요성 이해와 GS, SP인증과 함께 해외 인증 제시바랍니다.

| 문 제 | 7. CAP 이론의 한계와 PACELC 이론 | | |
|---|---|---|---|
| 출 제 영 역 | DB | 난 이 도 | ★★★☆☆ |
| 출 제 배 경 | - 분산데이터베이스의 CAP 이론 한계점을 보완하기 위해 나온 PACELC 이론에 대한 개념 및 관계 숙지 필요 | | |
| 출 제 빈 도 | 미출제 | | |
| 참 고 자 료 | - http://happinessoncode.com/2017/07/29/cap-theorem-and-pacelc-theorem/<br>- https://itwiki.kr/w/PACELC_%EC%9D%B4%EB%A1%A0 | | |
| K e y w o r d | -NoSQL, BASE, ACID | | |
| 풀 이 | 김유리(124 회 정보관리기술사) | | |

## 1. NoSQL 의 기본 이론, CAP 이론의 개요




| 정의 |
|---|
| - 분산 데이터베이스 시스템의 세 가지 속성인 일관성(Consistency), 가용성(Availability), 파티션 허용성(Partition tolerance)을 나타내며, 이중 두가지 특성만 가능하다는 이론 |

- CAP 이론에서는 CP 와 AP 만 만족할 수 있지만 완벽한 CP 시스템과 완벽한 AP 시스템 사이에는 수많은 가능성이 있는 만큼 이론의 한계가 존재함

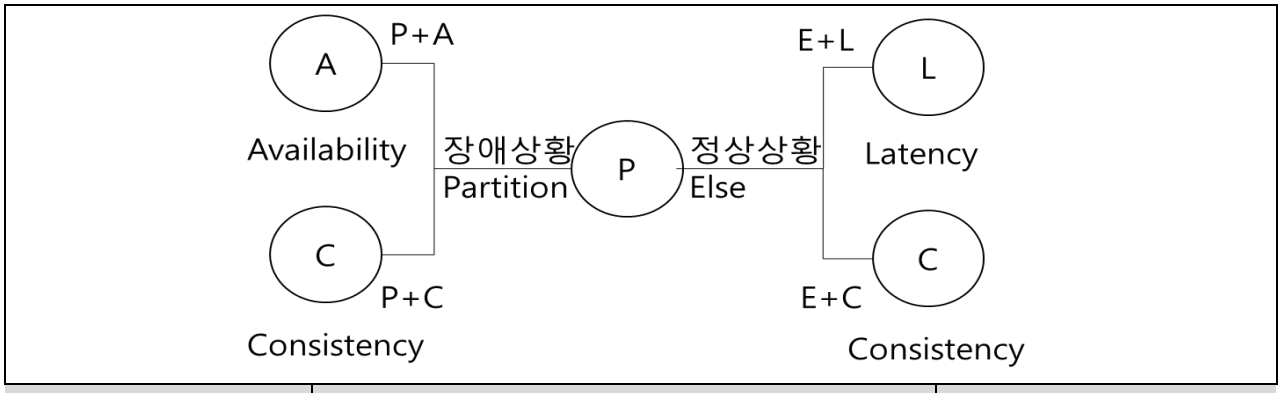
## 2. CAP 이론의 한계와 PACELC 이론의 설명

### 가. CAP 이론의 한계의 설명

| 한계 | 설명 |
|---|---|
| **① 완벽한 CP, AP 시스템은 사용할 수 없다** | - 완벽한 CP 시스템과 완벽한 AP 시스템 사이에는 수많은 가능성이 있기 때문에 일관성과 가용성은 상충 관계에 있지만 둘 중에 반드시 하나만을 선택해야 하는 것은 아니다 |
| **② 대부분 분산 시스템은 CP 와 AP 의 중간 어디쯤이다** | - 요구 사항에 따라 '다소 강한 일관성-다소 약한 가용성', '다소 약한 일관성-다소 강한 가용성'과 같이 일관성과 가용성의 수준을 선택이 필요함 |
| **③ 모든 분산 시스템이 파티션을 사용하진 않는다** | - 파티션이 없는 상황에서도 분산 시스템은 상충하는 특성들이 있고, 장애상황만큼 정상 상황에서 시스템이 어떻게 동작하는지도 중요함 |

- 기존 CAP 이론의 한계를 극복위해 네트워크 장애/정상 상황으로 나누어서 설명하는 PACELC 이론 등장

나. PACELC 이론의 설명




| 구분 | | 설명 | 대표 NoSQL |
|---|---|---|---|
| 장애 상황 | 정상 상황 | | |
| PC | EC | - 장애 상황 시 일관성 우선 보장<br>- 정상 상황 시 모든 노드 동일 메시지 보장 | - HBase, VoltDB, Megastore |
| PA | EL | - 장애 상황 시 가용 노드만 반영, 복구 시 전체 반영<br>- 정상 상황 시 Latency 우선 고려 | - Cassandra, Dynamo |
| PA | EC | - 장애 상황 시 가용 노드만 반영, 복구 시 전체 반영<br>- 정상 상황 시 모든 노드 동일 메시지 보장 | - MongoDB |
| PC | EL | - 장애 상황 시 Timeline Consistency 수준 보장<br>- 정상 상황 시 Latency 우선 고려 | - PNUTS |

- 데이터 베이스를 선택할 땐 CAP, PACELC 이론에 맞춰 비즈니스에 가장 적합한 솔루션을 선택하며 데이터 형태와 데이터 양에 따른 고려도 필요함

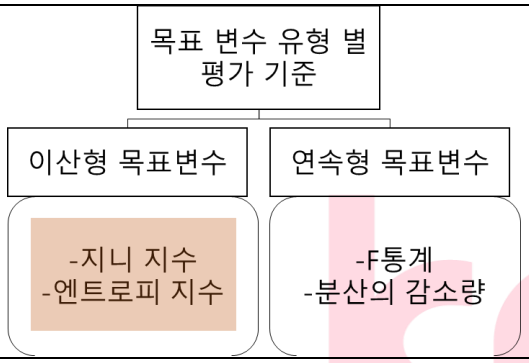
"끝"

**기출풀이 의견**

7. 분산DB의 CAP 이론의 개념과 이론의 한계점 및 해결을 위해 등장한 PACELC 이론을 연관 지어 설명 해주시면 됩니다. 추가로 PACELC 분류에 따라 사용되는 대표 NOSQL까지 작성해주시기 바랍니다.

| 문 제 | 8. 의사결정나무의 지니 지수(Gini Index)와 엔트로피 지수(Entropy Index) | | |
|---|---|---|---|
| 출 제 영 역 | 인공지능 | 난 이 도 | ★★★★☆ |
| 출 제 배 경 | - 빈번하게 출제된 데이터 분석 기법에 이어 심화된 알고리즘이 출제 | | |
| 출 제 빈 도 | 105 회 정보관리 | | |
| 참 고 자 료 | - https://velog.io/@noooooh_042/의사결정나무<br>-Ch06_01.R 의사결정나무(II)(지니,엔트로피,정보이익)01 - YouTube | | |
| Keyword | - 이산형 목표변수, 불순도, 순수도, CART, C4.5, CHAID | | |
| 풀 이 | 김유리(124 회 정보관리기술사) | | |

## 1. 분류(classification)와 회귀(regression) 분석 기법, 의사결정나무의 개요



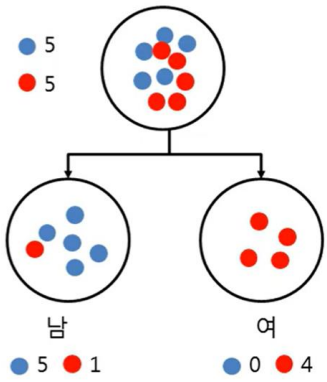

**정의**

- 모형의 구축과정을 나무형태로 표현하여 대상이 되는 집단을 몇 개의 소집단으로 구분하는 분류 및 예측 기법.

- 의사 결정 나무의 형성에 있어서 분석의 목적과 자료의 구조에 따라 적절한 분리기준(Split Criterion)과 정지규칙(Stopping Rules)을 정의
- 엔트로피, 지니 지수의 불순도 수치화한 지표를 사용하여 측정 자식마디의 순수도를 가장 높이는 변수를 분리 기준으로 결정함

## 2. 분류나무 불순도 측정, 지니 지수와 엔트로피 지수의 설명

### 가. 지니 지수(Gini Index)의 설명




※ 성별에 따른 지니 지수

$$G(\text{상위}) = 1 - \left(\frac{5}{10}\right)^2 - \left(\frac{5}{10}\right)^2 = 0.5$$

$$G(\text{남}) = 1 - \left(\frac{5}{6}\right)^2 - \left(\frac{1}{6}\right)^2 = 0.278$$

$$G(\text{여}) = 1 - \left(\frac{0}{4}\right)^2 - \left(\frac{4}{4}\right)^2 = 0$$

$$G(\text{성별}) = \left(\frac{6}{10}\right)(0.278) + \left(\frac{4}{10}\right)(0) = 0.167$$

∴ 부모 노드 불순도 0.5->0.167 감소로 불확실성의 감소로 성별 분리의 정확도가 높음

※ 지니 지수 수식

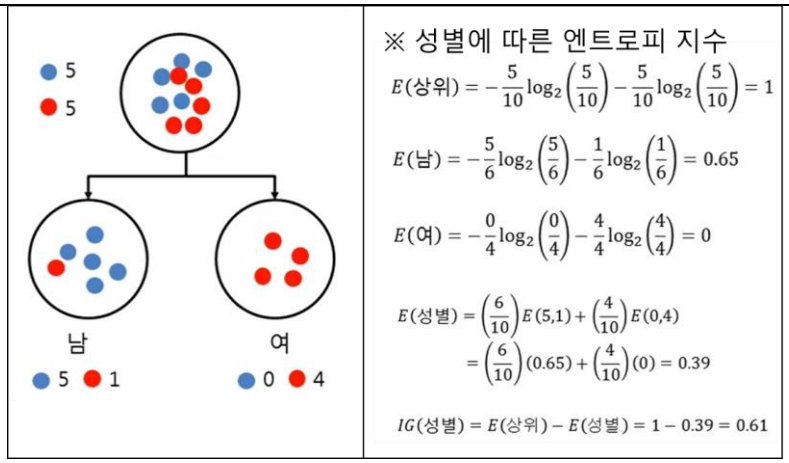
$$G(A) = 1 - \sum_{k=0}^{n} {p_k}^2$$

A: 이미 발생한 사건의 모음,
n: 사건의 개수

| 개념 | -CART 에서 사용하는 불순도 알고리즘으로 복원추출 개념을 이용하여 집합내에 이질적인 것이 얼마나 섞였는지를 측정하는 지표 |
|---|---|

| | |
|---|---|
| **설명** | -집합에 있는 항목이 모두 같다면 지니 불순도는 최솟값(0)<br>-지니 지수가 높을수록 노드 내의 이질성이 크고 순수도가 낮음<br>-지니 지수가 낮을수록 같은 특성을 가진 객체들로 분류가 잘 된 상태로 판단 |

**나. 엔트로피 지수(Entropy Index)의 설명**




※ 성별에 따른 엔트로피 지수

$$E(\text{상위}) = -\frac{5}{10}\log_2\left(\frac{5}{10}\right) - \frac{5}{10}\log_2\left(\frac{5}{10}\right) = 1$$

$$E(\text{남}) = -\frac{5}{6}\log_2\left(\frac{5}{6}\right) - \frac{1}{6}\log_2\left(\frac{1}{6}\right) = 0.65$$

$$E(\text{여}) = -\frac{0}{4}\log_2\left(\frac{0}{4}\right) - \frac{4}{4}\log_2\left(\frac{4}{4}\right) = 0$$

$$E(\text{성별}) = \left(\frac{6}{10}\right)E(5,1) + \left(\frac{4}{10}\right)E(0,4) = \left(\frac{6}{10}\right)(0.65) + \left(\frac{4}{10}\right)(0) = 0.39$$

$$IG(\text{성별}) = E(\text{상위}) - E(\text{성별}) = 1 - 0.39 = 0.61$$

※ 엔트로피 지수 수식

$$E(A) = -\sum_{k=1}^{m} p_k \log_2(p_k)$$

∴ 부모 노드 불순도 1->0.39 감소로 불확실성의 감소로 성별 분리의 정확도가 높음
- IG(정보이득) = 0.61로 값이 클수록 변별력이 좋음을 의미

| | |
|---|---|
| **개념** | -C4.5 에서 사용하는 불순도 알고리즘으로 log 를 사용하여 집합의 무질서한 정도를 측정하는 지표 |
| **설명** | -지수가 1 에 가까울 경우: 불순도가 높음, 0 에 가까울 경우: 불순도가 낮음<br>-지니 지수와 마찬가지로 불순도와 엔트로피(Entropy)는 비례관계 |

- 의사결정 나무는 다양한 재귀적 분할 알고리즘을 적용하며 그에 따른 불순도 알고리즘을 선택함

### 3. 재귀적 분할 알고리즘에 따른 비교

| 구분 | CART | C4.5 | CHAID |
|---|---|---|---|
| **분류나무(분류)** | ○ | ○ | ○ |
| **회귀나무(예측** | ○ | ○ | X |
| **예측변수** | 범주, 수치 | 범주, 수치 | 범주 |
| **불순도 알고리즘** | 지니 지수 | 엔트로피지수, GI | Chi-square 통계량 |
| **분리** | 이지분리 | 다지분리 | 다지분리 |
| **나무성장(멈추기)** | 완전모형 개발 | 완전보형 개발 | 최적모형 개발 |
| **가지치기(교차검정)** | 학습데이터>검증데이터 | 학습데이터 | X |
| **개발자** | Breiman | Quinlan | |

- 데이터 마이닝 기법 중 적용이 쉽고 해석이 편리한 의사결정나무를 보다 정확한 알고리즘의 기반 분석이 중요. "끝"

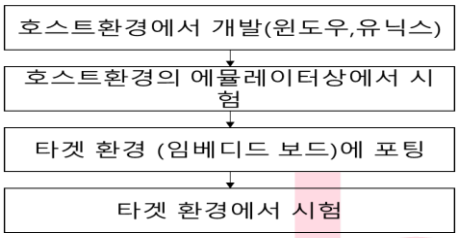
**기출풀이 의견**

8. 데이터 분석 기법에서 심화된 알고리즘이 출제된 만큼 분석 기법 별 주요 알고리즘의 개념 및 차이점등의 숙지 필요

| 문 제 | 9. 임베디드 소프트웨어 테스트(embedded software test) | | |
|---|---|---|---|
| 출 제 영 역 | 소프트웨어공학 | 난 이 도 | ★★★☆☆ |
| 출 제 배 경 | -임베디드 시스템의 개발 및 보급이 급속도로 발달함에 따라 그에 맞는 품질관리의 중요성 증가 | | |
| 출 제 빈 도 | 미출제 | | |
| 참 고 자 료 | - download.blog (tistory.com) | | |
| K e y w o r d | - 반응성, 실시간성, 프로세서, 입출력, 제어장치, JTAG, 에뮬레이터, 원격 디버깅 | | |
| 풀 이 | 김유리(124 회 정보관리기술사) | | |

## 1. 임베디드 SW 품질향상 전략, 임베디드 소프트웨어 테스트(embedded software test)의 개요

<임베디드 SW 개발 &테스트 과정>



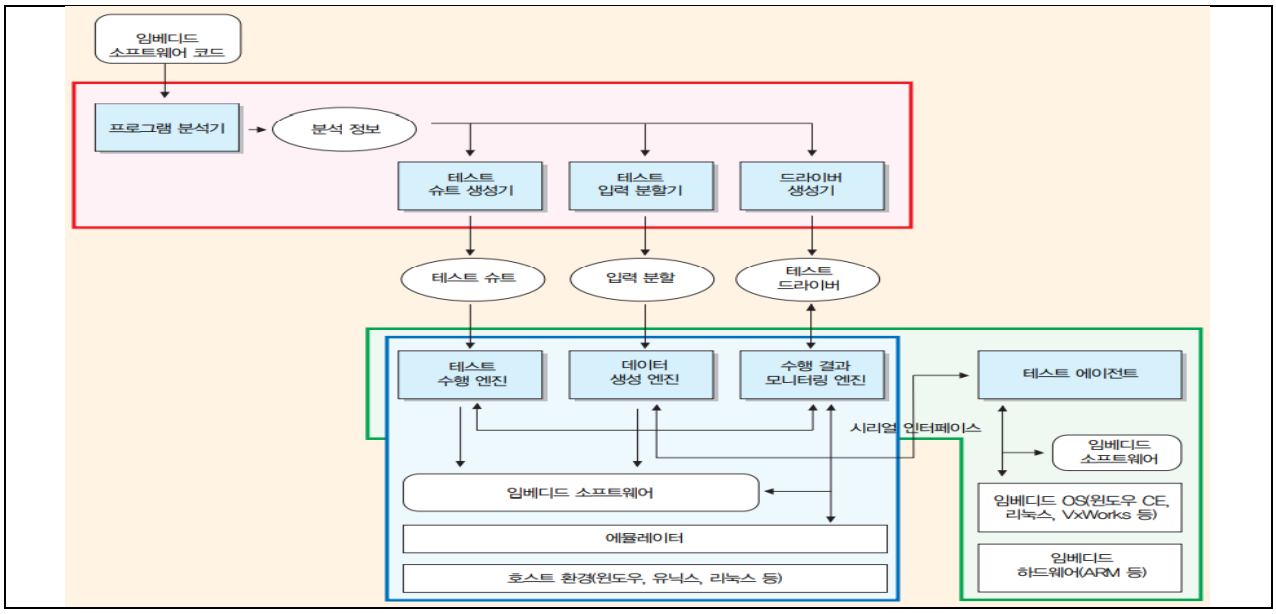

**정의**

- 임베디드 시스템에서 소프트웨어와 하드웨어의 기능적 및 비기능적 속성을 검사하고 최종 제품에 결함이 없는지 확인하기 위한 테스트 프로세스

- 임베디드 소프트웨어 테스팅은 일반 소프트웨어와 달리 개발환경(호스트 환경)과 운용환경(타깃 환경)이 다르므로 효과적인 테스팅을 위해서 기존 테스팅 도구를 임베디드 환경에 맞게 자동적 변경, 확장이 필요함

## 2. 임베디드 소프트웨어 테스트 자동화 아키텍처 및 설명

### 가. 임베디드 소프트웨어 테스트 자동화의 아키텍처




- 임베디드 시스템 테스트는 호스트 환경에서의 시험과 타겟 환경에서의 시험을 지원할 수 있어야 하며 특성상 HW, SW 동시 개발 진행으로 확정되지 않은 상태에서의 SW 테스팅이 요구되어져 코드 기반 테스트이 필요

나. 임베디드 소프트웨어 테스트 자동화 아키텍처 설명

| 구분 | 기능 | 설명 |
|---|---|---|
| 코드 기반 테스트 케이스 생성 부 | -화이트 박스 테스팅 기능 제공 | - 테스트 슈트 생성, 테스트 데이터 생성, 테스트 수행, 테스트 모니터링 및 결과 분석 |
| 호스트 환경시험 수행 부 | -에뮬레이터에서 기능 시험 지원<br>-호스트 개발환경 및 에뮬레이터와 결합<br>-test script/data/driver 자동 생성<br>-테스트 자동 수행 및 모니터링 | - 하드웨어가 확정되지 않은 상태에 서 소프트웨어 테스팅이 요구<br>- 개발 호스트 환경 및 타깃 환경에서의 시험 위해 테스팅 자동화 도구 간의 인터페이스를 제공으로 호스트 환경에서의 기능 시험 자동화를 지원 |
| 타깃 환경시험 수행 부 | -타깃 보드에서 기능 시험 지원<br>-타깃 OS 와 시리얼/USB 등으로 결합<br>-test script/data/driver 재사용<br>-test agent 를 통한 자동수행 및 모니터링 | -리소스에 대한 제약이 심해 테스트 도구가 타 깃 환경에 탑재된 상태의 테스트 진행은 어려움<br>-테스트 도구는 호스트 환경에서 수행하고, 테스팅 진행을 도울 수 있는 test agent 를 타깃 환경에 탑재해 테스팅을 진행 |

- 도구 자체가 임베디드 소프트웨어로 필수적으로 임베디드 환경에서 동작 해야 하는 테스트 케이스가 존재할 때 제약사항을 고려하여 개발이 필요함

## 3. 임베디드 시스템 소프트웨어 테스팅 이슈와 테스팅 시 고려해야할 사항

| 임베디드 소프트웨어 테스팅 이슈 | | 테스팅시 고려해야 할 사항 | |
|---|---|---|---|
| Event-driven | user-driven 기능과 event-driven 기능이 혼재됨. | Communication Media | 시리얼 라인 등 저사양인 경우가 대부분 |
| Time Criticial | 때때로 시간 제약 사항이 있는 경우가 존재함. | Monitoring Media | JTAG 등 모니터링을 위한 별도의 장비가 없는 경우가 많음. |
| Platform Diversity | 플랫폼(하드웨어, OS 등)이 매우 다양함. | Memory | 메모리 제약이 극심함. |
| Platform Stability | 플랫폼 자체에 오류가 있는 경우가 있음. | Hard Disk | 하드 디스크를 거의 사용하지 않음. |
| Development Env. | 컴파일러, 라이브러리 등이 불안정한 경우가 있음. | OS Facilities | 타이머, 메모리 보호 등 OS 기본 기능이 취약함. |

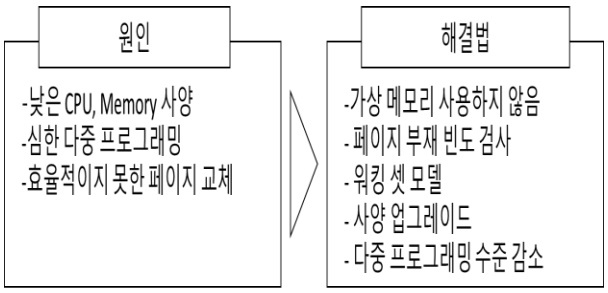
- Test Quest Pro, Tornado, CodeScroll 등 임베디드 특화된 테스트 도구가 많지는 않은 실정으로 IoT, 텔레매틱스 분야를 비롯한 다양한 분야에서 임베디드 소프트웨어 테스트의 중요성이 커지는 만큼 임베디드 소프트웨어 테스트 자동화 도구 개발 위한 노력이 필요함 "끝"

**기출풀이 의견**

9. 임베디드 시스템의 개발 및 보급이 급속도로 확산되고 있으나 임베디드 특화의 테스트 도구가 많지 않은 실정으로 임베디드 소프트웨어 테스트의 개념을 이해하고 현업에서의 이슈사항 및 해결책을 제시해주세요.

| 문 제 | 10. 스레싱(Thrashing) | | |
|---|---|---|---|
| 출 제 영 역 | CA/OS | 난 이 도 | ★★★☆☆ |
| 출 제 배 경 | - CPU 가 프로세스 실행보다 페이지 교체에 더 많은 시간을 소요하는 비정상 현상인 스레싱의 원인과 해결방안을 제시하는 기본 토픽 출제 | | |
| 출 제 빈 도 | 110 회 정보관리 | | |
| 참 고 자 료 | - https://itwiki.kr/w/%EC%8A%A4%EB%A0%88%EC%8B%B1<br>- https://zangzangs.tistory.com/144<br>-https://slidesplayer.org/slide/17747882/ | | |
| K e y w o r d | -지역성, 가상메모리, 페이지교체, Page Fault, Working set | | |
| 풀 이 | 김유리(124 회 정보관리기술사) | | |

## 1. 페이지 부재가 과도하게 발생하는 상황, 스레싱(Thrashing)의 개요




| 정의 |
|---|
| - 다중 프로그래밍과 가상 메모리 사용 환경에서 프로세스가 집중적으로 사용하는 페이지들의 집합이 메모리에 한꺼번에 적재되지 못하여 페이지 부재율(page fault)가 많이 발생하게 되고 CPU 이용율이 급격히 떨어지는 현상 |

- 다중 프로그래밍 정도가 높아짐에 따라 CPU의 이용율도 함께 높아지나, 특정지점 이후 페이지 교체 시간의 증가로 CPU 이용율이 낮아지는 현상 발생.

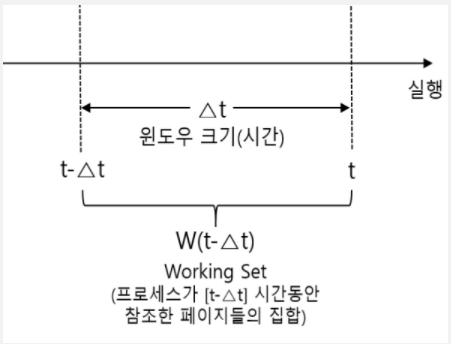
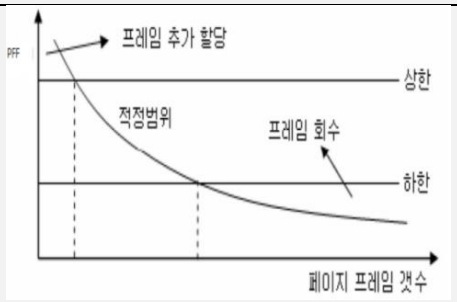
## 2. 스레싱(Thrashing)의 발생 원인 및 해결 방법의 설명

### 가. 스레싱(Thrashing)의 발생 원인의 설명

| 구분 | 발생원인 | 설명 |
|---|---|---|
| 자원부족 | - 저용량 Memory | - 메모리 부족으로 추가 리소스 적용 어려움 |
| | - 저사양 CPU | - 저사양의 CPU 사용으로 가용 자원 량이 부족 |
| 부적절한 페이지 교체 정책 | - Locality 미 고려 | - 공간, 시간 Locality 반영하지 못하여 부재율 증가 |
| | - 페이지빈도 미 고려 | - 페이지 교체에 대한 빈번한 시간 지연 문제 발생 |
| 과도한 멀티 프로그래밍 | - 할당 Frame 감소 | - 멀티 프로그램이 과도하여 할당 프레임이 제한됨 |
| | - Page Fault 증가 | - 프로세스의 수행에 필요한 페이지 프레임 할당이 어려워 페이지 부재가 빈번히 발생 |

- 대표적으로 지역성 고려통한 Working Set 기법이나 페이지 부재율 최소화의 PFF 통하여 스레싱을 예방할 수 있음
-

나. 스레싱(Thrashing)의 해결 방법

| 해결 방법 | 개념도 | 설명 |
|---|---|---|
| **Working Set Model** |  | – 지역성 집합이 메모리에 동시에 올라갈 수 있도록 보장하는 메모리 관리 알고리즘<br>- 매번 기억장치를 참고하고 난 뒤 페이지 집합 수정으로 Overhead 가 큼<br>– Working set: 메모리에 상주시킬 페이지 집합<br>– Working set window: page reference 의 고정된 숫자<br>– Working set size: working set 에 들어갈 최대 페이지 수 |
| **PFF (Page Fault Frequency)** |  | – Page Fault 발생시 Frame 수 조정 방식<br>– Page Fault 발생 시 에만 Frame 수 조정으로 Working set 비해 Overhead 가 낮음<br>- PFF>상한 값 (빈도 증가): Frame 추가 할당<br>- PFF<하한 값 (빈도 감소): Frame 회수 |

- 다중프로그래밍 수준을 감소하고 사양을 업그레이드함으로 스레싱을 예방할 수 있으며 페이지 정책, 프로그램 설계 고려로 시스템 성능 향상 위한 노력이 필요함

## 3. 페이징 시스템의 효과적 실행을 위한 고려사항

| 고려사항 | 설명 |
|---|---|
| **- 대치 범위** | - 여러 프로세스가 제한된 수의 프레임을 사용하려면 할당 기준이 필요함 |
| **- 프리 페이징** | - 한 번에 가져올 수 있는 페이지 수와 어떤 페이지를 가져올지를 결정할 수 있는 경험적 알고리즘의 선택이 필요함 |
| **– 페이지 크기** | – 페이지 수와 페이지 테이블의 크기, 내부단편화, 지역성 등의 전반적 고려사항에 포함 필요 |
| **- 페이지 테이블 구조** | - 일반적으로 각 프로세스는 하나의 페이지 테이블을 가짐, 페이지 테이블 역시 페이지의 대상이 되므로 글로벌 페이지 테이블의 고려 필요 |

- 페이징 시스템의 효과적 실행을 위해 실시간 처리, 입출력 상호잠금 등의 고려도 필요함

"끝"

**기출풀이 의견**

10. CPU 가 프로세스 실행보다 페이지 교체에 더 많은 시간을 소요하는 비정상 현상인 스레싱의 이해를 하고 원인과 해결방안을 다양하게 제시하시기 바랍니다.

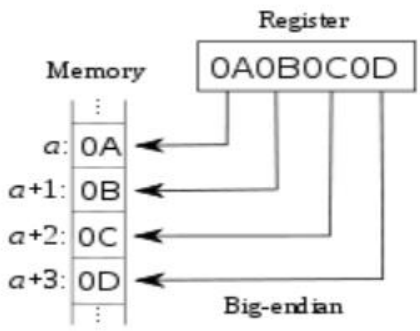
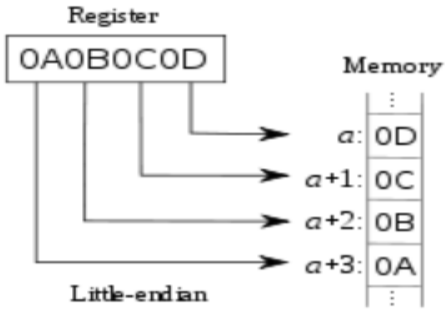
| 문 제 | 11. Big Endian 과 Little Endian 비교 | | |
|---|---|---|---|
| 출 제 영 역 | CA/OS | 난 이 도 | ★★★☆☆ |
| 출 제 배 경 | - 메모리 저장 방식의 유형에 대한 기본적 지식 요구 | | |
| 출 제 빈 도 | 미출제 | | |
| 참 고 자 료 | - http://soen.kr/lecture/ccpp/cpp2/18-1-3.htm<br>- https://ko.wikipedia.org/wiki/엔디언<br>- https://bigenergy.tistory.com/entry/C-TCP-Socket-통신시-빅엔디안과-리틀엔디안-변환방법 | | |
| Keyword | -바이트 저장 순서, MSB, LSB | | |
| 풀 이 | 김유리(124 회 정보관리기술사) | | |

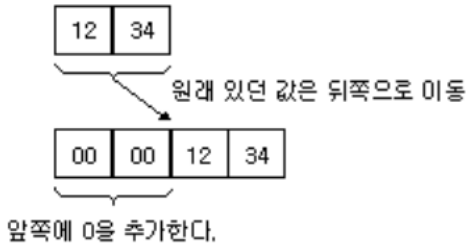
## 1. 바이트 저장 순서에 따른 분류, Big Endian 과 Little Endian 의 개념 비교

| Big Endian | Little Endian |
|---|---|
| - 최상위 바이트(MSB-Most Significant Byte) 부터 차례로 저장하는 바이트 배열 방식 | - 최하위 바이트(LSB-Least Significant Byte) 부터 차례로 저장하는 바이트 배열 방식 |

- X86아키텍처 사용의 대부분 데스크탑은 리틀에디언을, 네트워크 주소는 빅에디언으로 사용

## 2. Big Endian 과 Little Endian 의 상세 설명

| 구분 | Big Endian | Little Endian |
|---|---|---|
| 개념도 |  |  |
| 저장 방식 | 0x1234의 표현: 12 34<br>0x123456678의 표현: 12 34 56 78<br>-메모리의 0 에서부터 끝으로 쓰는 방식 | 0x1234의 표현: 34 12<br>0x123456678의 표현: 78 56 34 12<br>-메모리의 끝에서부터 0 으로 쓰는 방식 |

| | | |
|---|---|---|
| **확장 방식** |  |  |
| | -확장 시 여분의 연산이 필요함 | - 임시적인 축소나 확장에 유연함 |
| **장점** | -SW 디버깅 과정의 편의 제공<br>-비교 연산에서 속도 우수<br>-자연스러운 방식으로 가독성 우수 | -하위 바이트만 사용하는 계산에 용이<br>-계산 연산 성능 우수<br>-기계가 값을 다룰 경우 효율적 |
| **단점** | -계산의 복잡함<br>-형변환시 속도 저하 | -바이트를 배열처럼 다룰 때 불편함 |
| **적용** | - RISC 프로세서 계열 시스템<br>-모토로라 계열의 CPU<br>-소켓 프로그래밍의 네트워크 바이트 오더 | - Intel 프로세서 계열 시스템<br>- DEC 의 알파 프로세서<br>-연산에 최적화 |

- 값의 조각을 저장하는 순서가 다른 것뿐이며 CPU 설계자들은 CPU 의 구조나 설계 방식, 활용 방안 등에 따라 두 방식 중 하나를 선택한 것뿐임

- 윈도우계열의 리틀 엔디언과 서버군인 빅엔디안 간의 소켓 통식의 경우 정렬방법 차이로 문제 발생 가능해 네트워크 오더링으로 엔디언 변환작업이 필요

### 3. 윈도우와 유닉스 통신시 엔디언 변환의 사례

| 구분 | 예시 | |
|---|---|---|
| **윈도우 송신** | -<br>Arry.Copy(BitConverter.GetBytes(IPAddress.HostToNetworkOrder((short)"변수")),0,"바이트배열", idx,2); | -빅엔디안으로 데이터 정렬 |
| **윈도우 수신** | - IPAddress. NetworkToHostOrder() | -리틀엔디안으로 변환 |

- c 언어에서 _big_endian 키워드나 REV/REV16/REVSH 명령을 사용하며 엔디언간 변환 가능

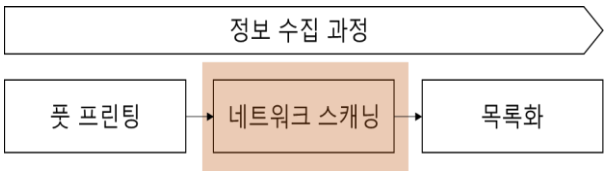
"끝"

**기출풀이 의견**

11. 메모리 저장 방식의 유형에 대한 기본적 지식을 요구하는 문제로 유형 별로 MSB, LSB 부터 저장하는 방식의 차이를 비교설명으로 작성 해주시기 바랍니다.

| 문 제 | 12. 네트워크 스캐닝(Network Scanning) | | |
|---|---|---|---|
| 출 제 영 역 | 보안 | 난 이 도 | ★★★☆☆ |
| 출 제 배 경 | - 최근 로그 4 셸(Log4Shell) 보안 취약점 발견 이슈 배경 | | |
| 출 제 빈 도 | 미출제 | | |
| 참 고 자 료 | - https://tkdrms568.tistory.com/78<br>- https://www.boannews.com/media/view.asp?idx=103589 | | |
| K e y w o r d | -포트 스캐닝, 풋 프린팅, 목록화 | | |
| 풀 이 | 김유리(124 회 정보관리기술사) | | |

## 1. 사이버 공격을 위한 정보 수집, 네트워크 스캐닝(Network Scanning)의 개요

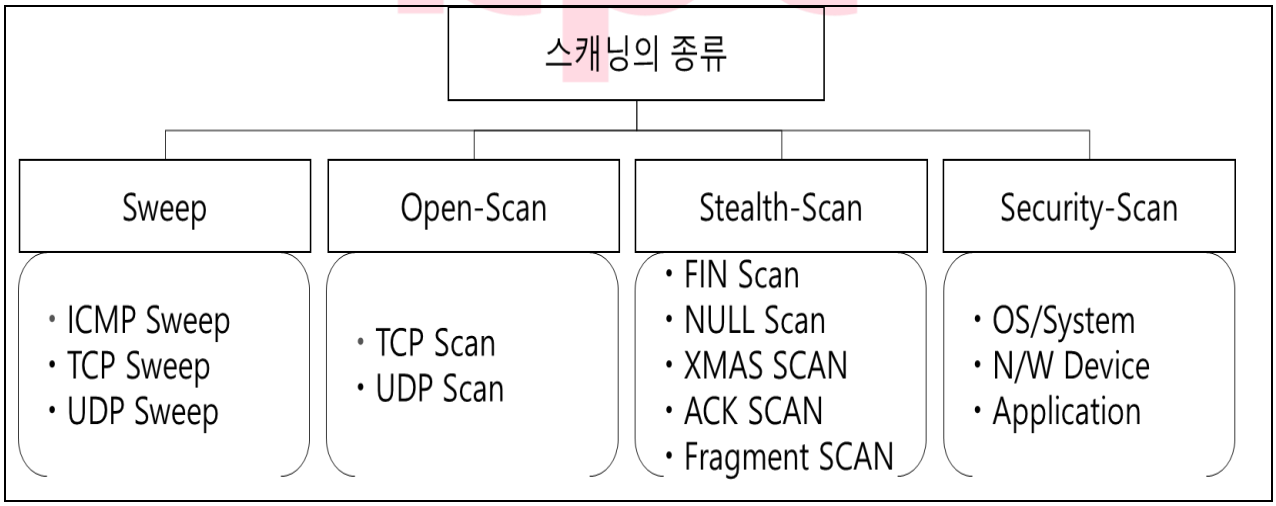



| 정의 |
|---|
| - 공격에 필요한 정보를 얻거나 이용 가능한 서비스와 포트를 조사 및 취약점 검색을 위한 정보 획득의 과정. |

- 실제 공격방법 결정 또는 공격에 이용될 수 있는 네트워크 구조, 시스템이 제공하는 정보를 얻는 활동

## 2. 네트워크 스캐닝(Network Scanning)의 종류

### 가. 네트워크 스캐닝(Network Scanning)의 종류




- 네트워크 스캐닝은 데이터 패킷이 지정된 포트 번호로 전송되는 포트 스캐닝을 의미하기도 함

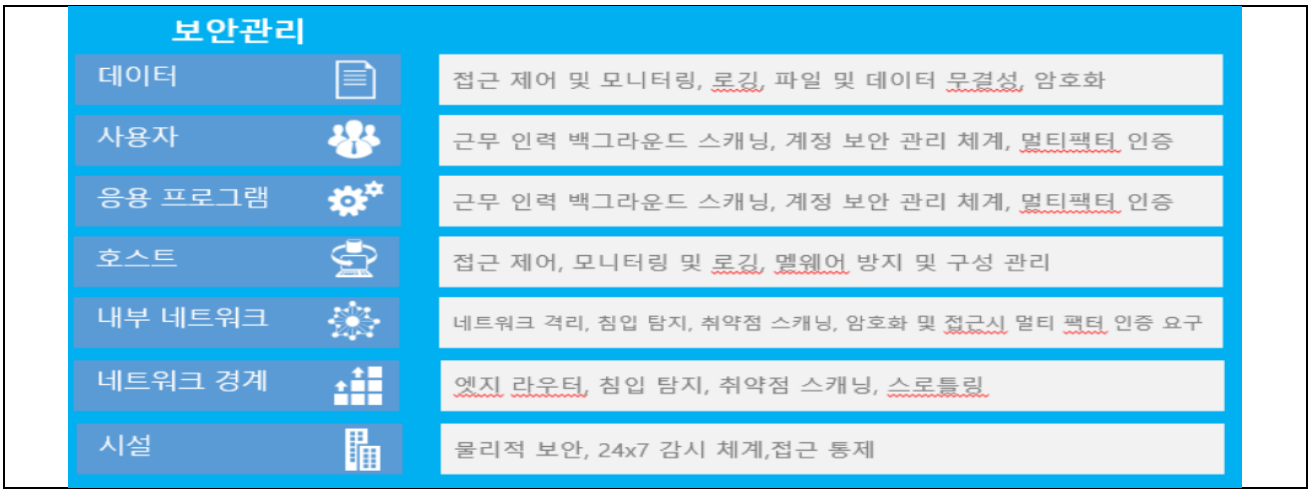
### 나. 대표 네트워크 스캐닝의 상세 설명

| 종류 | 상세 기법 | 설명 |
|---|---|---|
| Sweep | - N/W 에 속해 있는 System 유/무 판단, 타겟 기관의 IP 주소 및 N/W 범위 파악<br>- 서버/클라이언트 구조 | |
| | -ICMP Sweep (Ping Sweep) | -소규모 네트워크 환경에 신속하게 시스템 동작여부 |

| | | |
|---|---|---|
| | | 점검 |
| **Open Scan** | - System 자체 활성화 여부, 스캔하는 포트의 서비스 활성화 여부 확인 | |
| | TCP Scan<br>TCP Full Open Scan<br>-TCP SYN/ACK Scan<br>TCP Half Open Scan<br>-TCP SYN Scan | - 종류에 따라 완전한 연결 시 TCP Full-Open, 완전한 연결 아니면 TCP Half-Open 으로 나눌 수 있음<br>-Full Open-Scan 은 TCP 이용해 신뢰성 있는 정보 습득은 가능하지만 속도가 느리고 로그가 남기 때문에 탐지가 가능<br>-Half-Open Scan 은 포트 활성화 여부 만 확인 |
| | -UDP Scan | - ICMP Port Unreachable 에러 패킷 수신 시 포트가 닫혀 있으며 아무 응답 없으면 포트가 열려 있는 것으로 판단 |
| **Stealth Scan** | -TCP 헤더를 조작해 특수 패킷을 만들어 전송 후 응답으로 포트 개방여부 확인 | |
| | -FIN-Scan | - TCP 헤더 내 FIN 플래그 설정해 메시지 전송 후 포트가 열려 있으면 무응답, 포트가 닫혀 있으면 RST 패킷 전송해 포트의 개방여부 확인 |
| | -Null-Scan | - TCP 헤더의 Flag 설정을 하지 않고 메시지 전송 방식이며 결과는 FIN-Scan 과 동일 |
| | -Xmas-Scan | - TCP 헤더 내 FIN, ACK, SYN, URG, RST 플래그를 모두 설정해 전송 방식이며 결과는 FIN-Scan 과 동일 |
| **NMap** | nmap [scan_type] [port_option] [target] | - 모든 운영체제에서 사용 가능<br>- 운영체제 종류, 사용서비스에 대한 정보, FTP 서버의 취약점 이용한 bounce 공격 수행 가능한 스캔 도구 |

- 포트 스캔은 단순한 조사이기 때문에 실제 손해가 발생하진 않지만 이로 인한 취약점 발견 시 정보 유출 가능성이 크므로 피해에 대응마련이 필요함

### 3. 네트워크 스캔 공격의 대응책




- 사이버 공격자들이 활발히 네트워크 스캔을 함으로써 log4j 쉘 취약점과 파생 취약점을 노리는 등의 공격이 끊이지 않고 있으므로 SW, 시스템 뿐 아니라 네트워크 자동화를 정착시켜 보안 강화 위한 대응 전략이 필요

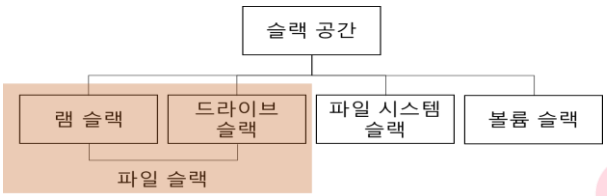
"끝"

**기출풀이 의견**

12. 네트워크 스캐닝을 통해 로그4쉘과 그 외 취약점 검색에 사용되고 있는 만큼 개념 및 유형에 대한 이해가 필요하며 대응마련까지 제시 해주시기 바랍니다

| 문 제 | 13. 파일 슬랙(File Slack) | | |
|---|---|---|---|
| 출 제 영 역 | CA/OS | 난 이 도 | ★★★☆☆ |
| 출 제 배 경 | - 디지털 환경 확산에 따라 범죄 수집 증거에 사용되는 디지털 포렌식 기법을 출제 | | |
| 출 제 빈 도 | 미출제 | | |
| 참 고 자 료 | - http://forensic.korea.ac.kr/DFWIKI/index.php/Slack<br>-슬랙 공간 (Slack Space Area) \| FORENSIC-PROOF | | |
| K e y w o r d | -디지털 포렌식, 슬랙 공간, 램/드라이브/파일시스템/볼륨 슬랙 | | |
| 풀 이 | 김유리(124 회 정보관리기술사) | | |

## 1. 디지털 포렌식 주요 분석 기법, 파일 슬랙(File Slack)의 개요



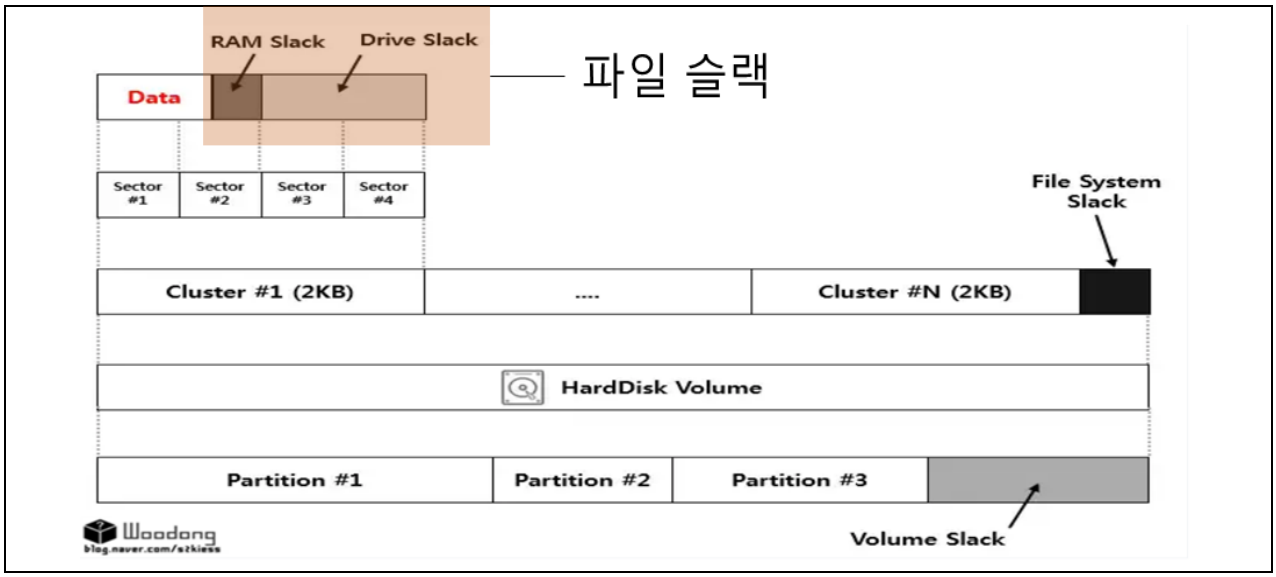

**정의**

- 저장매체의 물리적인 구조와 논리적인 구조의 차이로 발생하는 공간으로 램 슬랙과 드라이브 슬랙을 합친 공간

- 섹터와 클러스터로 나눠 파일을 저장하는 곳으로 할당하는 과장에서 자연스럽게 낭비되는 공간이 슬랙 공간
- 슬랙 공간은 사용자와 운영체제 의해 접근 될 수 없기 때문에 완전히 제거가 어렵단 속성에 기반해 디지털 포렌식에서 중요하게 사용되는 기술

## 2. 슬랙 공간(Slack Space Area)의 개념도 및 파일 슬랙(File Slack)의 설명

### 가. 슬랙 공간(Slack Space Area)의 개념도




- 램 슬랙과 드라이브 슬랙 모두 파일에 기반하여 나타나는 슬랙이기 때문에 파일 슬랙으로 불림

나. 파일 슬랙(File Slack)의 설명

| 구분 | 특징 | 설명 |
|---|---|---|
| 램 슬랙(RAM Slack) | – 지정되는 파일 크기가 512 바이트의 배수가 아닐 경우 발생하는 공간<br>– 섹터 슬래그로도 불리며 데이터가 기록된 가장 마지막 섹터의 남는 공간 | – 1 섹터 단위로 저장<br>– eof(end of file) 구분을 위해 남는 공간은 0x00 으로 채워짐 |
| 드라이브 슬랙(Drive Slack) | - 하나의 클러스터 단위에서 데이터와 램 슬랙을 제외한 공간<br>-0x00 채워지는 램 슬랙과 달리 아무것도 하지 않는 공간 | -데이터가 있다면 초기화 시 오버헤드의 원인의 소지가 존재함<br>-고의로 악성코드 기록의 소지가 있는 공간 |

- 그 외 슬랙 공간으로 파일시스템이 할당 후 남은 자리인 파일시스템 슬랙과 하드디스크 볼륨의 파티션을 나누고 남은 공간인 볼륨 슬랙이 존재함
- 포렌식 기술 중 파일을 복구 기법인 파일 카빙에서 램 슬랙의 특성중 0 x00 으로 채워지는 성질을 이용한 램슬랙 카빙이 있음

## 3. 디지털 포렌식 파일 복구 기법 파일 카빙의 개요

| 구분 | 연속적 카빙<br>(Continuous Carving) | 비연속적 카빙<br>(Fragment Recovery Carving) |
|---|---|---|
| 개념 | -데이터가 저장매체의 연속된 공간에 저장된 경우 수행하는 기법 | -데이터 단편화가 발생해 장매체의 여러 부분에 조각나 저장된 경우 수행하는 기법 |
| 개념도 |  |  |
| 기법 | -헤더/부터 카빙, 램슬랙 카빙, 파일 크기 카빙, 파일 검증 카빙 | -시그니처 탐색, 엔트로피, 바이트 분포, 바이트 편차 |
| **개념** | - 일반 파일, DBMS 페이지, 문서 스트림, 특정 데이터 구조, 문자열 등의 바이너리 데이터로부터 의미 있는 정보를 획득하는 기법<br>- 파일의 메타 정보가 덮어써진 경우 파일을 복구하는 방법 | |

- 슬랙 공간에 정보 은닉, 파일 복구 및 삭제된 파일의 조사 시 유용하게 사용할 수 있어 포렌식 분석 시 고려되어져야 함

"끝"

**기출풀이 의견**

13. 디지털 환경 확산에 따라 범죄 수집 증거에 기여할 디지털 포렌식 기법 출제로 슬랙 공간의 개념 및 종류를 제시하고 그 중 파일슬랙에 대한 제시가 필요합니다. 디지털 포렌식관점에서의 파일 슬랙의 분석 방법을 제시하시면 더 좋은 점수를 받을 수 있을 것입니다.

-