살충제 패러독스

동일한 테스트 케이스에 의한 반복적 테스트는 새로운 버그를 찾지 못한다는 테스트의 원리

데이터 마이닝

대규모로 저장된 데이터 안에서 체계적이고 자동적으로 통계적 규칙이나 패턴을 찾아내는 기술

프로토콜의 3 요소

1. 구문 - 시스템 간의 정보 전송을 위한 데이터 형식, 코딩, 신호 레벨 등을 규정
2. 의미 - 시스템 간의 정보 전송을 위한 제어 정보로 조정과 에러 처리를 위한 규정
3. 타이밍 - 시스템 간의 정보 전송을 위한 속도 조절과 순서 관리 규정

(프로토콜은 **복수의 컴퓨터 사이에서 데이터 통신을 원활하게 하기 위해 필요한 통신규약**입니다. 대표적으로 흔히 사용되는 IP/TCP 가 있습니다.)

JSON

비동기 브라우저. 서버 통신(AJAX)을 위해 '속성-값 쌍', '키-값-쌍'으로 이루어진 데이터 오브젝트를 전달하기 위해 인간이 읽을 수 있는 텍스트를 사용하는 개방형 표준 포맷

비선점형 스케줄링 알고리즘 유형

우선순위(Priority) - 프로세스별로 우선순위가 주어지고, 우선순위에 따라 CPU 를 할당

기한부(Deadine) - 작업들이 명시된 시간이나 기한 내에 완료되도록 계획

FCFS(First Come First Service) - 프로세스가 대기 큐에 도착한 순서에 따라 CPU 를 할당

SJF(Shortest Job First) - 프로세스가 도착하는 시점에 따라 그 당시 가장 작은 서비스 기간을 갖는 프로세스가 종료까지 원 점유

HRN(Highest Response Ratio Next) - 대기 중인 프로세스 중 현재 응답률이 가장 높은 것을 선택

(대기 시간 + 서비스 시간) / 서비스 시간

(비선점형이란 **하나의 프로세스가 끝나지 않으면 다른 프로세스는 CPU 를 사용할 수 없다** 란 뜻입니다.)

선점형 스케줄링 알고리즘 유형

RR(Round Robin) - 시분할 시스템에서 사용 / FCFS 와 비슷하지만 제한시간이 지난 후, 다음 프로세스에게 자원을 할당

SRT(Shortest Remaining Time) - SJF 기법을 선점형으로 바꾼 스케줄링

선점 우선순위 - 비선점 우선순위를 선점형으로 바꾼 것

다단계 큐 - 프로세스의 우선순위에 따라 시스템 프로세스, 대화형 프로세스, 일괄처리 프로세스 등으로 나누어 준비상태 큐를 상위 중위 하위단계로 배치

다단계 피드백 큐 - 다단계 큐의 단점을 보완함 / 큐마다 timeout 을 설정하여 timeout 초과시 우선순위가 낮은 다음단계 큐로 이동

(선점형이란 **하나의 프로세스가 다른 프로세스 대신에 프로세서(CPU)를 차지할 수 있다** 란 뜻입니다.)

XML

송.수신 시스템 간 데이터 연계의 편의성을 위해서 전송되는 데이터 구조를 동일한 형태로 정의

트랜잭션 특성

원자성 - 분해가 불가능한 작업의 최소단위

일관성 - 트랜잭션이 실행 성공 후 항상 일관된 데이터베이스 상태를 보존해야하는 특성

격리성 - 트랜잭션 실행 중 생성하는 연산의 중간 결과를 다른 트랜잭션이 접근 불가한 특성

영속성 - 성공이 완료된 트랜잭션의 결과는 영속적으로 데이터베이스에 저장하는 특성

(트랜잭션이란 데이터베이스의 상태를 변화시키는 하나의 논리적 기능을 수행하기 위한 작업의 단위 또는 한꺼번에 모두 수행되어야 할 일련의 연산을 의미합니다.)

TCL(Transaction Control Language)의 명령어

커밋 - 트랜잭션 확정 트랜잭션을 메모리에 영구적으로 저장하는 명령어

롤백 - 트랜잭션 취소 트랜잭션 내역을 저장 무효화시키는 명령어

체크 포인트 - 저장 시기 설정 롤백을 위한 시점을 지정하는 명령어

랜드 어택

출발지(Source) IP 와 목적지(Destination) IP 를 같은 패킷 주소로 만들어 보냄으로써 수신자가 자기 자신에게 응답을 보내게 하여 시스템의 가용성을 침해하는 공격 기법이다

입력 데이터 검증 및 표현에 대한 취약점

XSS (Cross Site Script) - 검증되지 않은 외부 입력 데이터가 포함된 웹 페이지가 전송되는 경우, 사용자가 해당 웹 페이지를 열람함으로써 웹 페이지에 포함된 부적절한 스크립트가 실행되는 공격

사이트 간 요청 위조 (CSRF) - 사용자가 자신의 의지와는 무관하게 공격자가 의도한 행위를 특정 웹 사이트에 요청하게 하는 공격

SQL 삽입 (Injection) - 응용 프로그램의 보안 취약점을 이용해서 악의적인 sql 구문을 삽입, 실행시켜서 데이터베이스의 접근을 통해 정보를 탈취하거나 조작 등을 행위하는 공격 기법

해시 암호화 알고리즘 종류

MD5 (Message-Digest algorithm 5) - 각각의 512bit 짜리 입력 메세지 블록에 대해 차례로 동작하여 128bit 의 해시값을 생성하는 해시 알고리즘

SHA-1 (Secure Hash Algorithm) - 160bit 의 해시값을 생성하는 해시 알고리즘

SHA-256/384/512 (Secure Hash Algorithm) - SHA 알고리즘의 한 종류로써 256bit 의 해시값을 생성하는 해시 함수

HAS-160 - 국내 표준 서명 알고리즘 KCDSA(Korean Certificate-based Digital Signature Algorithm)를 위하여 개발된 해시 함수

애플리케이션 성능 측정 지표

처리량 - 애플리케이션이 주어진 시간에 처리할 수 있는 트랜잭션의 수

응답시간 - 사용자 입력이 끝난 후, 애플리케이션의 응답 출력이 개시될때까지의 시간

경과 시간 - 애플리케이션에 사용자가 요구를 입력한 시점부터 트랜잭션을 처리 후 그 결과의 출력이 완료할 때까지 걸리는 시간

자원 사용률 - 애플리케이션이 트랜잭션을 처리하는 동안 사용하는 CPU 사용량, 메모리 사용량, 네트워크 사용량

응집도의 유형

우연적 응집도 - 모듈 내부의 각 구성요소가 연관이 없을 경우의 응집도

논리적 응집도 - 유사한 성격을 갖거나 특정 형태로 분류되는 처리 요소들이 한 모듈에서 처리되는 경우의 응집도

시간적 응집도 - 연관된 기능이라기보다는 특정 시간에 처리되어야 하는 활동들을 한 모듈에서 처리할 경우의 응집도

절차적 응집도 - 모듈 다수의 관련 기능을 가질 때 모듈 안의 구성요소들이 그 기능을 순차적으로 수행할 경우의 응집도

통신적 응집도 - 동일한 입력과 출력을 사용하여 다른 기능을 수행하는 활동들이 모여 있을 경우의 응집도

순차적 응집도 - 모듈 내에서 한 활동으로부터 나온 출력값을 다른 활동이 사용할 경우의 응집도

기능적 응집도 - 모듈 내부의 모든 기능이 단일한 목적을 위해 수행되는 경우의 응집도

응집도 나즘(나쁜 품질)에서 응집도 높음(좋은 품질)의 순서

우연적 응집도 - 논리적 응집도 - 시간적 응집도 - 절차적 응집도 - 통신적 응집도 - 순차적 응집도 - 기능적 응집도

(응집도는 **모듈에 포함된 내부 요소들이 하나의 책임 / 목적을 위해 연결되어 있는 연관된 정도**입니다.)

결합도의 유형

내용 결합도 - 다른 모듈 내부에 있는 변수나 기능을 다른 모듈에서 사용하는 경우의 결합도

공통 결합도 - 파라미터가 아닌 모듈 밖에 선언되어 있는 전역 변수를 참조하고, 전역 변수를 갱신하는 식으로 상호작용하는 경우의 결합도

외부 결합도 - 두 개의 모듈이 외부에서 도입된 데이터 포맷, 통신 프로토콜, 또는 디바이스 인터페이스를 공유할 경우의 결합도

제어 결합도 - 단순 처리할 대상인 값만 전달되는 게 아니라 어떻게 처리를 해야 한다는 제어 요소가 전달되는 경우의 결합도

스탬프 결합도 - 모듈 간의 인터페이스로 배열이나 객체, 구조 등이 전달되는 경우의 결합도

자료 결합도 - 모듈 간의 인터페이스로 전달되는 파라미터를 통해서만 모듈간의 상호작용이 일어나는 경우의 결합도

결합도 높음(낮은 품질)에서 결합도 낮음(좋음 품질)의 순서

내용 결합도 - 공통 결합도 - 외부 결합도 - 제어 결합도 - 스탬프 결합도 - 자료 결합도

(결합도란 다른 모듈과의 의존성이 정도입니다.)

소프트웨어 개발방법론 종류

구조적 방법론 - 전체 시스템을 기능에 따라 나누어 개발하고, 이를 통합하는 분할과 정복 접근 방식의 방법론

정보공학 방법론 - 정보시스템 개발에 필요한 관리 절차와 작업 기반을 체계화한 방법론

개체지향 방법론 - '객체'라는 기본 단위로 시스템을 분석 및 설계하는 방법론

컴포넌트 기반 방법론 - 소프트웨어를 구성하는 컴포넌트를 조립해서 하나의 새로운 응용 프로그램을 작성하는 방법론

애자일 방법론 - 절차보다는 사람이 중심이 되어 변화에 유연하고 신속하게 적용하면서 효율적으로 시스템을 개발할 수 있는 신속 적응적 경량 개발방법론

제품 계열 방법론 - 특정 제품에 적용하고 싶은 공통된 기능을 정의하여 개발하는 방법론

릴리즈 노트의 주요 작성 항목

해더 - 문서이름, 제품이름 , 버전 번호, 릴리즈 날짜, 참고 날짜, 노트 버전 등의 정보

개요 - 제품 및 변경에 대한 간략한 전반적 개요

목적 - 릴리스 버전의 새로운 기능목록과 릴리스 노트의 목적에 대한 개요, 버그 수정 및 새로운 기능 기술

이슈 요약 - 버그의 간단한 설명 또는 릴리즈 추가 항목 요약

재현 항목 - 버그 발견에 따른 재현 단계 기술

개선 내용 - 개선의 간단한 설명 기술

사용자 영향도 - 버전 변경에 따른 최종 사용자 기준의 기능 및 응용 프로그램상의 영향도 기술

소프트웨어 지원 영향도 - 버전 변경에 따른 소프트웨어의 지원 프로세스 및 영향도 기술

노트 - 소프트웨어 및 하드웨어 설치 항목, 제품 문서를 포함한 업그레이드 항목 메모

면책 조항 - 회사 및 표준 제품과 관련된 메세지, 프리웨어 및 불법 복제 방지, 중복 등 참조에 대한 고지 사항

연락 정보 - 사용자 지원 및 문의에 관련한 연락처 정보

(릴리스 노트는 소프트웨어 제품과 함께 배포되는 문서들을 말합니다.)

반정규화의 주요 기법

테이블 병합 - 1:1 관계, 1:M 관계를 통합하여 조인 횟수를 줄여 성능을 향상

테이블 분할 - 테이블을 수직 또는 수평으로 분할하는 것으로 파티셔닝이라고 함

중복 테이블 추가 - 대량의 데이터들에 대한 집계 함수(group by, sum 등)를 사용하여 실시간 통계 정보를 계산하는 경우에 효과적인 수행을 위해 별도의 통계 테이블을 두거나 중복 테이블을 추가

컬럼 중복화 - 조인 성능 향상을 위해 중복 허용

중복 관계 추가 - 데이터를 처리하기 위한 여러 경로를 거쳐 조인이 가능하지만, 이때 발생할 수 있는 성능 저하를 예방하기 위해 추가적 관계를 맺는 방법

(반정규화란 **시스템의 성능 향상, 개발 및 운영의 편의성 등을 위해 정규화된 데이터 모델을 통합, 중복, 분리하는 과정으로 의도적으로 정규화 원칙을 위배하는 행위**입니다.)

OSI 7 계층의 특징

응용 계층 - 사용자와 네트워크 간 응용서비스 연결, 데이터 생성

표현 계층 - 데이터 형식 설정, 부호교환, 암.복호화

세션 계층 - 송수신 간의 논리적 연결 / 연결 접속, 동기제어

전송 계층 - 송수신 프로세스 간의 연결 / 신뢰성 있는 통신 보장 / 데이터 분할, 제조립, 흐름제어, 오류 제어, 혼잡 제어

네트워크 계층 - 단말기 간 데이터를 전송하기 위한 최적화된 경로 제공

데이터링크 계층 - 인접 시스템 간 데이터 전송, 전송 오류 제어 / 동기화, 오류 제어, 흐름 제어, 회선 제어

물리 계층 - 0 과 1 의 비트 정보를 회선에 보내기 위한 전기적 신호 변환

(OSI 7 계층이란 **국제 표준화 기구인 ISO(Internatioal Standardization Organization)에서 개발한 컴퓨터 네트워크 프로토콜 디자인과 통신을 계층으로 나누어 설명한 개방형 시스템 상호 연결 모델**입니다.)

AJAX(Asynchronous JavaScript and XML) 비동기 통신 기법

- 브라우저가 가지고 있는 XMLHttpRequest 객체를 이용해서 전체 페이지를 새로 고치지 않고도 페이지 일부만을 위한 데이터를 로드하는 기법
- 하이퍼텍스트 표기 언어(HTML)만으로 어려운 다양한 작업을 웹 페이지에서 구현해 이용자가 웹 페이지와 자유롭게 상호작용할 수 있도록 하는 기술

비즈니스 연속성 계획(BCP)의 주요 용어

BIA(Business Impact Analysis) - 장애나 재해로 인해 운영상의 주요 손실을 볼 것을 가정하여 시간 흐름에 따른 영향도 및 손실평가를 조사하는 BCP 를 구축하기 위한 비즈니스 영향 분석

RTO(Recovery Time Objective) - 업무중단 시점부터 업무가 복구되어 다시 가동될 때까지의 시간

RPO(Recovery Point Objective) - 업무중단 시점부터 데이터가 복구되어 다시 정상가동될 때 데이터의 손실 허용 시점

DRP(Disaster Recovery Plan) - 재난으로 장기간에 걸쳐 시설의 운영이 불가능한 경우를 대비한 재난 복구 계획

DRS(Disaster Recovery System) - 재해복구계획의 원활한 수행을 지원하기 위하여 평상시에 확보하여 두는 인적,물적 자원 및 이들에 대한 지속적인 관리체계가 통합된 재해복구센터

(비즈니스 연속성 계획(BCP)이란 **각종 재해나 재난발생에 대비하여 핵심 업무 기능수행의 연속성을 유지**하여 고객 서비스의 지속성 보장과 고객에 대한 신뢰도를 높이는 신속한 절차와 체계를 구축해 기업의 가치를 최대화 해주는 방법론)

IPSec 의 주요 프로토콜

인증(AH) 프로토콜 - 메세지 인증 코드(MAC)를 이용하여 인증과 송신처 인증을 제공해주는 프로토콜로 기밀성(암호화)은 제공하지 않는 프로토콜

암호화(ESP) 프로토콜 - 메세지 인증 코드(MAC)와 암호화를 이용하여 인증과 송신처 인증과 기밀성을 제공하는 프로토콜

키 관리(IKE) 프로토콜 - Key 를 주고 받는 알고리즘 / 공개된 네트워크를 통하여 Key 를 어떻게 할 것인가를 정의

(IPSec 는 IP 계층(3 계층)에서 무결성과 인증을 보장하는 인증 헤더(AH)와 기밀성을 보장하는 암호화(ESP)를 이용한 IP 보안 프로토콜입니다.)

디자인 패턴 중 행위 패턴

Mediator - 객체지향 설계에서 객체 수가 많아지면 서로 간 통신을 위해 복잡해져서 객체지향에서 가장 중요한 느슨한 결합의 특성을 해칠 수 있기에 중간에서 이를 통제하고 지시할 수 있는 역할의 중재자를 두고, 중재자에게 모든 것을 요구하여 통신의 빈도를 줄여 객체지향의 목표를 달성하게 해줌

Interpreter - 언어의 다양한 해석, 구체적으로 구문을 나누고, 그 분리된 구문의 해석을 맡는 클래스를 각각 작성하여 여러 형태의 언어 구문을 해석할 수 있게 만듬

Iterator - 컬렉션 구현 방법을 노출시키지 않으면서도 그 집합체 안에 들어가있는 모든 항목에 접근할 방법을 제공

Template Method - 어떤 작업을 처리하는 일부분을 서브 클래스로 캡슐화해 전체 일을 수행하는 구조는 바꾸지 않으면서 특정 단계에서 수행하는 내역을 바꿈

Observer - 한 객체의 상태가 바뀌면 그 객체에 의존하는 다른 객체들에 연락이 가고 자동으로 내용이 갱신되는 방법

State - 객체 상태를 캡슐화하여 클래스화함으로써 그것을 참조하게 하는 방식

Visitor - 각 클래스 데이터 구조로부터 처리 기능을 분리하여 별도의 클래스를 만들어 놓고 해당 클래스의 메서드가 각 클래스를 돌아다니며 특정 작업을 수행하도록 만듬

Command - 실행될 기능을 캡슐화함으로써 주어진 여러 기능을 실행할 수 있는 재사용이 높은 클래스를 설계

Strategy - 알고리즘 군을 정의하고(추상 클래스) 같은 알고리즘을 각각 하나의 클래스로 캡슐화한 다음, 필요할 때 서로 교환해서 사용할 수 있게 하는 패턴

Memento - 클래스 설계 관점에서 객체의 정보를 저장할 필요가 있을 때 적용하는 디자인 패턴

Chain of Responsibility - 정적으로 어떤 기능에 대한 처리의 연결이 하드 코딩되어 있을 때 기능처리의 연결 변경이 불가능한데, 이를 동적으로 연결된 경우에 따라 다르게 처리될 수 있도록 연결한 디자인 패턴

(디자인 패턴이란 소프트웨어를 설계할 때 특정 맥락에서 자주 발생하는 고질적인 문제들이 또 발생했을 때 재사용할 수 있는 훌륭한 해결책을 말합니다. 디자인패턴에는 생성패턴 / 구조패턴 / 행위패턴이 있습니다.

생성패턴 - 객체 생성에 관련된 패턴

구조패턴 - 클래스나 객체를 조합해 더 큰 구조를 만드는 패턴

행위패턴 - 객체나 클래스 사이의 알고리즘이나 책임 분배에 관련된 패턴)

안드로이드의 특징

리눅스 기반 - 안드로이드는 리눅스 커널 위에서 동작

자바와 코틀린 언어 - 고수준 언어를 사용해 응용 프로그램을 작성

런타임 라이브러리 - 컴파일된 바이트 코드 구동 기능

안드로이드 소프트웨어 개발 - 응용 프로그램을 개발하는데 필요한 각종 도구와 API 를 제공

(안드로이드는 구글에서 개발한 운영체제로 리눅스 위에서 구동하며, 휴대폰 전화를 비롯한 휴대용 장치를 위한 운영체제와 미들웨어, 사용자 인터페이스 그리고 표준 응용 프로그램(웹 브라우저 등) 등을 포함하고 있는 소프트웨어 스택이자 리눅스 모바일 운영체제입니다.)

SOAP(Simple Object Access Protocol)

- SOAP 는 HTTP, HTTPS, SMTP 등을 사용하여 XML 기반의 메세지를 네트워크 상태에서 교환하는 프로토콜

UI 설계 원칙

직관성(Intuitiveness) - 누구나 쉽게 이해하고, 쉽게 사용할 수 있어야 함

유효성(Effciency) - 정확하고 완벽하게 사용자의 목표가 달성될 수 있도록 제작

학습성(Learnability) - 초보와 숙련자 모두가 쉽게 배우고 사용할 수 있게 제작

유연성(Flexibility) - 사용자의 인터랙션을 최대한 포용하고, 실수를 방지할 수 있도록 제작

LOD(Linked Open Data)

웹상에 존재하는 데이터를 개별 RUI(Uniform Resource Identifier)로 식별하고, 각 URI 에 링크 정보를 부여함으로써 상호 연결된 웹을 지향하는 데이터

데이터모델링의 절차

개념적 데이터 모델 - 현실 세계에 대한 인식을 추상적, 개념적으로 표현하여 개념적 구조를 도출하는 데이터 모델

논리적 데이터 모델 - 업무의 모습을 모델링 표기법으로 형상화화여 사람이 이해하기 쉽게 표현한 데이터 모델

물리적 모델 - 논리 데이터 모델을 특정 DBMS 의 특성 및 성능을 고려하여 물리적인 스키마를 만드는 일련의 모델

형상 관리의 절차

형상 식별 - 형상 관리 대상을 정의 및 식별하는 활동

형상 통제 - 형상 항목의 버전 관리를 위한 형상통제위원회 운영

형상 감사 - 소프트웨어 베이스라인의 무결성 평가

형상 기록 - 소프트웨어 형상 및 변경 관리에 대한 각종 수행결과를 기록

(형상 관리는 소프트웨어 개발을 위한 전체 과정에서 발생하는 모든 항목의 변경 사항을 관리하기 위한 활동입니다.)

리팩토링의 목적

유지보수성 향상 - 복잡한 코드의 단순화, 소스의 가독성 향상

유연한 시스템 - 소프트웨어 요구사항 변경에 유연한 대응

생산성 향상 - 정제 및 최적화된 소스의 재사용

품질 향상 - 소프트웨어 오류발견이 용이하여 품질향상

(리팩토링은 소프트웨어 모듈의 외부적 기능은 수정하지 않고 내부적으로 구조, 관계 등을 단순화하여 소프트웨어의 유지보수성을 향상시키는 기법입니다.)

OSPF 의 특징

다익스트라 알고리즘 사용 - 다익스트라 알고리즘을 사용하는 내부 라우팅 프로토콜

라우팅 메트릭 지정 - 최조, 지연 , 최대 처리량 등 관리자가 라우팅 메트릭 지정

AS 분할 사용 - 자치 시스템을 지역으로 나누어 라우팅을 효과적으로 관리

홉 카운트 무제한 - 홉 카운트에 제한이 없음

(OSPF (Open Shortest Path First)는 대표적인 내부 라우팅 프로토콜로 다익스트라 알고리즘을 이용한 대규모 네트워크에 적합한 링크 상태 라우팅 프로토콜로도 불리는 라우팅 프로토콜 OSPF 입니다.)

ICMP(Internet Control Message Protocol)

IP 패킷을 처리할 때 발생하는 문제를 알려주는 프로토콜로, 메세지 형식은 8 비트의 헤더와 가변 길이의 데이터 영역으로 분리되어 있음

식별자 표기법

가멜 표기법 - 식별자 표기 시에 여러 단어가 이어지면 첫 단어 시작만 소문자로 표시하고, 각 단어의 첫 글자는 대문자로 지정하는 표기법 (ex : goodMan)

파스칼 표기법 - 식별자 표기 시에 여러 단어가 이어지면 각 단어의 첫 글자는 대문자로 지정하는 표기법

(ex : GoodMan)

스네이크 표기법 - 식별자 표기 시에 여러 단어가 이어지면 단어 사이에 언더 바를 넣는 표기법 (ex : good_man)

헝가리안 표기법 - 식별자 표기 시, 두어에 자료형을 붙이는 표기법 (ex : goodMan > 정수형)

블랙박스 테스트의 유형

동등분할 테스트 - 입력 데이터의 영역을 유사한 도메인별로 유효 값/ 무효 값을 그룹핑하여 대푯값 테스트 케이스를 도출하여 테스트하는 기법

경곗값 분석 테스트 - 등가 분할 후 경곗값 부분에서 오류 발생 확률이 높기 때문에 경곗값을 포함하여 테스트 케이스를 설계하여 테스트하는 기법

결정 테이블 테스트 - 요구사항의 논리와 발생 조건을 테이블 형태로 나열하여 조건과 행위를 모두 조합하여 테스트하는 기법

상태 전이 테스트 - 테스트 대상, 시스템이나 객체의 상태를 구분하고 이벤트에 의해 어느 한 상태에서 다른 상태로 전이되는 경우의 수를 수행하는 테스트 기법

유스케이스 테스트 - 시스템이 실제 사용되는 유스케이스로 모델링 되어있을 때 프로세스 흐름을 기반으로 테스트 케이스를 명세화하여 수행하는 테스트 기법

분류 트리 테스트 - SW 의 일부 또는 전체를 트리 구조로 분석 및 표현하여 테스트 케이스를 설계하여 테스트하는 기법

페어와이즈 테스트 - 테스트 데이터값 간에 최소한 한번씩 조합하는 방식

원인-결과 그래프 테스트 - 그래프를 활용하여 입력 데이터 간의 관계 및 출력에 미치는 영향을 분석하여 효용성이 높은 테스트 케이스를 선정하여 테스트하는 기법

비교 테스트 - 여러 버전의 프로그램에 같은 입력값을 넣어서 동일한 결과 데이터가 나오는지 비교해 보는 테스트 기법

(블랙박스 테스트는 소프트웨어 검사 방법 중 하나로 어떤 소프트웨어를 내부 구조나 작동 원리르 모르는 상태에서 소프트웨어의 동작을 검사하는 방법입니다.)

EAI 의 구축 유형

포인트 투 포인트 - 가장 기초적인 애플리케이션 통합방법으로 1:1 단순 통합방법

허브 앤 스포크 - 단일한 접점의 허브 시스템을 통하여 데이터를 전송하는 중앙 집중식 방식

메시지 버스 - 애플리케이션 사이 미들웨어(버스)를 두어 연계하는 미들웨어 통합 방식

하이브리드 - 그룹 내부는 허브 앤 스포크 방식을 사용하고, 그룹 간에는 메세지 버스 방식을 사용하는 통합 방식

(EAI(Enterprise Application Integration)란 기업 응용 프로그램의 구조적 통합 방안을 가리킵니다.)

IPv4 주소

- 주소체계는 10 진수로 총 12 자리이며, 네 부분으로 나뉜다.
- 각 부분은 0~255 까지 3 자리의 수로 표현된다.
- IPv4 주소는 32bit 로 구성되어 있으며, 인터넷 사용자의 증가로 인해 주소 공간의 고갈로 128bit 주소체계를 갖는 IPv6 가 등장하고 점점 확산되고 있다.

IPv6 주소

- IPv4 의 기존 32bit 주소 공간에서 벗어나, IPv6 는 128bit 주소 공간을 제공하고, IPv6 는 네트워크의 물리적 위치에 제한받지 않고 같은 주소를 유지한다.

디자인 패턴의 유형

목적

1. 생성 - 객체 인스턴스 생성에 관여, 클래스 정의와 객체 생성 방식을 구조화, 캡슐화를 수행하는 패턴
2. 구조 - 더 큰 구조 형성 목적으로 클래스나 객체의 조합을 다루는 패턴
3. 행위 - 클래스나 객체들이 상호작용하는 방법과 역할 분담을 다루는 패턴

범위

1. 클래스 - 클래스 간 관련성 / 컴파일 타임에 정적으로 결정
2. 객체 - 객체 간 관련성을 다루는 패턴 / 런타임에 동적으로 결정

네트워크 공격 기법

스니핑 - 공격 대상에게 직접 공격하지 않고 데이터만 몰래 들여다보는 수동적 공격 기법

네트워크 스캐너, 스피너 - 네트워크 하드웨어 및 소프트웨어 구성의 취약점 파악을 위해 공격자가 취약점을 탐색하는 공격 도구

패스워드 크래킹 - 사전 크래킹 공격, 무차별 크래킹 공격, 패스워드 하이브리드 공격, 레인보우 테이블 공격 활용

IP 스푸핑 - 침입자가 인증된 컴퓨팅 시스템인 것처럼 속여서 타깃 시스템의 정보를 빼내기 위해서 본인의 패킷 헤더를 인증된 호스트의 IP 어드레스로 위조하여 타깃에 전송하는 공격기법

ARP 스푸핑 - 공격자가 특정 호스트의 MAC 주소를 자신의 MAC 주소로 위조한 ARP Reply 를 만들어 희생자에게 지속적으로 전송하여 희생자의 ARP Cache Table 에 특정 호스트의 MAC 정보를 공격자의 MAC 정보로 변경, 희생자로부터 특정 호스트로 나가는 패킷을 공격자가 스니핑하는 공격 기법

ICMP Redirect 공격 - 3 계층에서 스니핑 시스템을 네트워크에 존재하는 또 다른 라우터라고 알림으로써 패킷의 흐름을 바꾸는 공격 기법

트로이 목마 - 악성 루틴이 숨어 있는 프로그램으로 겉보기에는 정상적인 프로그램으로 보이지만 실행하면 악성 코드를 실행하는 프로그램

NAT 유형

Static NAT - 사설 IP 주소와 공인 IP 주소가 1:1 로 연결되는 구성

Dynamic NAT - 사설 IP 와 공인 IP 주소가 N:1 또는 N:M 으로 연결되는 구성

(NAT(Network Address Transformation) 는 보통 사설 네트워크에 속한 여러 개의 호스트가 하나의 공인 IP 주소를 사용하여 인터넷에 접속하기 위하여 사용합니다.)

블록체인 합의 알고리즘

PoW (Proof of Work) - 확률적으로 해답이 어려운 문제를 가장 빨리 해결한 사람에게 블록을 만들 수 있도록 허가

PoS (Proof of Stake) - 이더리움이 채택한 알고리즘으로 화폐량을 더 많이 소유하고 있는 승인자가 우선하여 블록을 생성할 수 있는 알고리즘

(블록체인은 분산 컴퓨팅 기술 기반의 데이터 위변조 방지 기술로 P2P 방식을 기반으로하여 소규모 데이터들이 연결되어 형성된 '블록'이라는 분산 데이터 저장 환경에 관리 대상 데이터를 저장함으로써 누구도 임의로 수정할 수 없고 누구나 변경의 결과를 열람할 수 없게끔 만드는 기술입니다.)

하둡(Hadoop)의 구성

하둡 분산 파일 시스템 (HDFS) - 대용량 파일을 분산된 서버에 저장하고 그 저장된 데이터를 빠르게 처리할 수 있게 하는 시스템

맵리듀스 (Map Reduce) - 대용량 데이터 세트를 분산 병렬 컴퓨팅에서 처리하거나 생성하기 위한 목적으로 만들어진 소프트웨어 프레임워크

(하둡은 오픈 소스를 기반으로 한 분산 컴퓨팅 플랫폼으로, 일반 PC 급 컴퓨터들로 가상화된 대형 스토리지를 형성하고 그 안에 보관된 거대한 데이터 세트를 병렬로 처리할 수 있도록 개발된 자바 소프트웨어 프레임워크로 구글, 야후 등에 적용한 기술입니다.)

데이터베이스 이상 현상

삽입 이상 - 정보 저장 시 해당 정보의 불필요한 세부정보를 입력해야 하는 경우

삭제 이상 - 정보 삭제 시 원치 않는 다른 정보가 같이 삭제되는 경우

갱신 이상 - 중복 데이터 중에서 특정 부분만 수정되어 중복된 값이 모순을 일으키는 경우

(데이터베이스 이상 현상이란 갱신/삽입/삭제하였을때 그 속성의 다른 속성값들과의 불일치가 발생하는 현상입니다.)

프로세스 상태 전이

생성 상태 - 사용자에 의해 프로세스가 생성된 상태

준비 상태 - CPU 를 할당받을 수 있는 상태

실행 상태 - 프로세스가 CPU 를 할당받아 동작 중인 상태

대기 상태 - 프로세스 실행 중 입출력 처리 등으로 인해 CPU 를 양도하고 입출력 처리가 완료까지 대기 리스트에서 기다리는 상태

완료 상태 - 프로세스가 CPU 를 할당받아 주어진 시간 내에 완전히 수행을 종료한 상태

(프로세스 상태 전이는 다중 프로그래밍 환경을 바탕하기에 상태 전이가 이루어집니다.)

테스트 오라클 유형

참 오라클 - 모든 입력값에 대하여 기대하는 결과를 생성함으로써 발행된 오류를 모두 검출할 수 있는 오라클

샘플링 오라클 - 특정한 몇 개의 입력값에 대해서만 기대하는 결과를 제공해주는 오라클

휴리스틱 오라클 - 샘플링 오라클을 개선한 오라클로, 특정 입력값에 대해 올바른 결과를 제공하고, 나머지 값들에 대해서는 휴리스틱(추정)으로 처리하는 오라클

일관성 검사 오라클 - 애플리케이션 변경이 있을 때, 수행 전과 후의 결과값이 동일한지 확인하는 오라클

(테스트 오라클은 테스트의 결과가 참인지 거짓인지를 판단하기 위해서 사전에 정의된 참값을 입력하여 비교하는 기법입니다.)

서버 접근통제 유형

임의적 접근통제(DAC:Discretionary Access Control) - 시스템에 대한 접근을 사용자/그룹의 신분 기반으로 제한하는 방법

강제적 접근통제(MAC:Mandatory Access Control) - 시스템 정보의 허용등급을 기준으로 사용자가 갖는 접근 허가 권한에 근거하여 시스템에 대한 접근을 제한하는 방법

역할 기반 접근통제 (RBAC:Role Based Access Control) - 중앙 관리자가 사용자와 시스템의 상호관계를 통제하며 조직 내 맡은 역할에 기초하여 자원에 대한 접근을 제한하는 방법

네트워크 계층(3 계층) 프로토콜 종류

IP (Internet Protocol) - 송수신 간의 패킷 단위로 데이터를 교환하는 네트워크에서 정보를 주고받는 데 사용하는 통신 프로토콜

ARP (Address Resolution Protocol) - IP 네트워크상에서 IP 주소를 MAC 주소(물리 주소)로 변환하는 프로토콜

RARP (Reverse Address Resolution Protocol) - IP 호스트가 자신의 물리 네트워크 주소(MAC)는 알지만 IP 주소를 모르는 경우, 서버로부터 IP 주소를 요청하기 위해 사용하는 프로토콜

ICMP (Internet Control Message Protocol) - IP 패킷을 처리할 때 발생하는 문제를 알려주는 프로토콜 메세지 형식은 8bit 헤더와 가변 길이의 데이터 영역으로 분리

IGMP (Internet Group Management Protocol) - 인터넷 그룹 관리 프로토콜은 호스트 컴퓨터와 인접 라우터가 멀캐스트 그룹 멤버십을 구성하는데 사용하는 통신 프로토콜

라우팅 프로토콜 (Routing Protocol) - 데이터 전송을 위해 목적지까지 갈 수 있는 여러 경로 중 최적의 경로를 설정해 주는 라우터 간의 상호 통신 프로토콜

DB 설계 절차

요구사항 분석 - 사용자에게서 데이터베이스를 사용하는 용도를 파악

개념적 설계 - 요구사항 명세서를 기반으로 개념적 데이터 모델을 표현하며 E-R 다이어그램으로 표현할 수 있음

논리적 설계 - 목표 DBMS 에 맞는 스키마 설계, 트랜잭션 인터페이스를 설계하는 정규화 과정을 수행함

물리적 설계 - 특정 DBMS 의 특성 및 성능을 고려하여 데이터베이스 저장 구조로 변환하는 과정으로 결과로 나오는 명세서는 테이블 저의서 등이 있음

구현 - SQL 문을 실행하여 데이터베이스를 실제로 생성함

테스트 레벨 종류

단위 테스트 - 사용자 요구사항에 대한 단위 모듈, 서브루틴 등을 테스트하는 단계 (개발)

통합 테스트 - 단위 테스트를 통과한 모듈 사이의 인터페이스, 통합된 컴포넌트 간의 상호작용을 검증하는 테스트 단계 (설계)

시스템 테스트 - 통합된 단위 시스템의 기능이 시스템에서 정상적으로 수행되는지를 검증하는 테스트 단계 (기계 명세 분석)

인수 테스트 - 계약상의 요구사항이 만족하였는지 확인하기 위한 테스트 단계 (요구사항 분석)

알파 테스트 - 선택된 사용자가 개발자 환경에서 통제된 상태로 개발자와 함께 수행하는 인수 테스트

베타 테스트 - 실제 환경에서 일정 수의 사용자에게 대상 소프트웨어를 사용하게 하고 피드백을 받는 인수 테스트

회귀 테스트 - 회귀 테스트는 오류를 제거하거나 수정한 시스템에서 오류 제거와 수정 때문에 새로이 유입된 오류가 없는지 확인하는 일종의 반복 테스트 기법

IPC 기법

메시지 큐 - 메시지 단위로 동작하여 프로세스 간 통신함

공유메모리 - 한 프로세스의 일부분을 다른 프로세스와 공유

소켓 - 클라이언트와 서버 프로세스 둘 사이에 통신을 가능하게 함

세마포어 - 프로세스 사이의 동기를 맞추는 기능을 제공함

(IPC(Inter-Process Communication)는 프로세스 간 통신 기술입니다.)

데이터 모델 구성요소

연산 - 데이터베이스에서 저장된 실제 데이터를 처리하는 작업에 대한 명세

구조 - 논리적으로 표현된 개체 타입 간의 관계

제약조건 - 데이터베이스에 저장될 수 있는 실제 데이터의 논리적인 제약조건