

다이어그램

- 사물과 관계를 도형으로 표현

- 정적 모델링에서는 주로 구조적 다이어그램을 사용하고 동적 모델링에서는 주로 행위 다이어그램 사용

- 구조적 다이어그램

-> 클래스 다이어그램 : 클래스, 클래스가 가지는 속성, 클래스 사이 관계 표현

-> 객체 다이어그램 : 인스턴스를 특정 시점의 객체와 객체 사이의 관계로 표현

-> 컴포넌트 다이어그램 : 구현 단계에서 사용되며 컴포넌트 간의 관계나 인터페이스를 표현

-> 패치 다이어그램 : 구현단계에서 사용되며 결과물, 프로세스, 컴포넌트 등 물리적 요소들의 위치 표현

-> 복합체 구조 다이어그램 : 복잡한 구조를 가지는 클래스 혹은 컴포넌트의 내부 구조 표현

-> 패키지 다이어그램 : 유스케이스나 클래스 등의 모델 요소들을 그룹화한 패키지들의 관계 표현

- 행위 다이어그램

-> 유스케이스 다이어그램 : 사용자의 요구를 분석하여 기능 모델링 작업에 사용됨

-> 시퀀스 다이어그램 : 상호 작용하는 시스템이나 객체들이 주고받는 메시지 표현

-> 커뮤니케이션 다이어그램 : 객체들이 주고받는 메시지를 표현할 뿐 아니라 객체들 간의 연관까지 표현

-> 상태 다이어그램 : 하나의 객체가 자신이 속한 클래스의 상태 변화 혹은 다른 객체와의 상호 작용에 따라 어떻게 변화하는지 표현

-> 활동 다이어그램 : 객체의 처리 로직이나 조건에 따른 처리의 흐름을 순서에 따라 표현

-> 상호작용 개요 다이어그램 : 상호작용 다이어그램 간의 제어 흐름 표현

-> 타이밍 다이어그램 : 객체 상태 변화와 시간 제약을 명시적으로 표현

결합도

- 모듈 간에 상호 의존도 또는 모듈 사이의 연관 관계

- 결합도와 품질은 반비례 관계

- 결합도가 강하면 시스템 구현 및 유지보수 작업이 어려움

세기	종류	내용
강함	내용 결합도	<ul style="list-style-type: none"> 다른 모듈의 내부 기능 및 그 내부 자료를 직접 참조하거나 수정할 때의 결합도 다른 모듈의 내부로 제어가 이동하는 경우도 포함
	공통 결합도	<ul style="list-style-type: none"> 공유되는 공통 데이터 영역을 여러 모듈이 사용할 때의 결합도 공통 데이터 영역의 내용을 변경 시 이를 사용하는 모든 모듈에 영향을 미쳐 모듈의 독립성을 약하게 만듦
	외부 결합도	<ul style="list-style-type: none"> 모듈에서 선언한 데이터를 외부의 다른 모듈에서 참조할 때의 결합도 참조되는 데이터의 범위를 각 모듈에서 제한
	제어 결합도	<ul style="list-style-type: none"> 다른 모듈 내부의 논리적인 흐름을 제어하기 위해 제어 신호를 이용하여 통신하거나 제어 요소를 전달하는 결합도 권리 전도 현상 : 하위 모듈이 상위 모듈에게 처리 명령을 내려는 현상
	스탬프 결합도	<ul style="list-style-type: none"> 모듈 간의 인터페이스가 배열이나 레코드 등의 자료구조가 전달할 때의 결합도 두 모듈이 동일한 자료 구조를 조회
약함	자료 결합도	<ul style="list-style-type: none"> 모듈 간의 인터페이스가 자료 요소로만 구성될 때의 결합도 모듈은 호출 시 매개 변수나 인수로 데이터를 넘겨주고 처리 결과를 반환

응집도

- 정보 은닉 개념을 확장한 것으로 모듈의 내부 요소들의 서로 관련되어 있는 정도
- 모듈이 독립적인 기능으로 정의되어 있는 정도
- 응집도와 품질은 비례 관계

세기	종류	내용
강함	기능적 응집도	<ul style="list-style-type: none"> 모듈 내부의 모든 기능 요소들이 단일 문제와 연관되어 수행될 경우의 응집도
	순차적 응집도	<ul style="list-style-type: none"> 모듈 내 하나의 활동으로부터 나온 데이터를 그 다음 활동의 입력 데이터로 사용할 경우의 응집도
	통신적 응집도	<ul style="list-style-type: none"> 동일한 입력과 출력을 사용하여 서로 다른 기능을 수행하는 구성 요소들이 모였을 경우의 응집도
	절차적 응집도	<ul style="list-style-type: none"> 모듈이 다수의 관련 기능을 가질 때 모듈 안의 구성 요소들이 그 기능을 순차적으로 수행할 경우의 응집도
	시간적 응집도	<ul style="list-style-type: none"> 특정 시간에 처리되는 몇 개의 기능을 모아 하나의 모듈로 작성할 경우의 응집도
약함	논리적 응집도	<ul style="list-style-type: none"> 유사한 성격을 갖거나 특정 형태로 분류되는 처리 요소들로 하나의 모듈이 형성되는 경우의 응집도

논리적 응집도 다음 우연적 응집도

생성 패턴

- 객체의 생성과 참조 과정을 샘플화 하여 객체가 생성되거나 변경되어도 프로그램의 구조에 영향을 크게 받지 않도록 하여 프로그램의 유연성을 더해줌
- 추상 팩토리 : 구체적인 클래스에 의존하지 않고 인터페이스를 통해 서로 연관, 의존하는 객체들의 그룹으로 생성하여 추상적으로 표현
- 빌더 : 작게 분리된 인스턴스를 건축 하듯이 조합하여 객체 생성
- 팩토리 메소드 : 객체 생성을 서브 클래스에서 처리하도록 분리하여 캡슐화한 패턴

- 프로토타입 : 원본 객체를 복제하는 방법으로 객체를 생성하는 패턴
- 싱글톤 : 하나의 객체를 생성하면 생성된 객체를 어디서든 참조할 수 있지만 여러 프로세스가 동시에 참조할 수는 없음

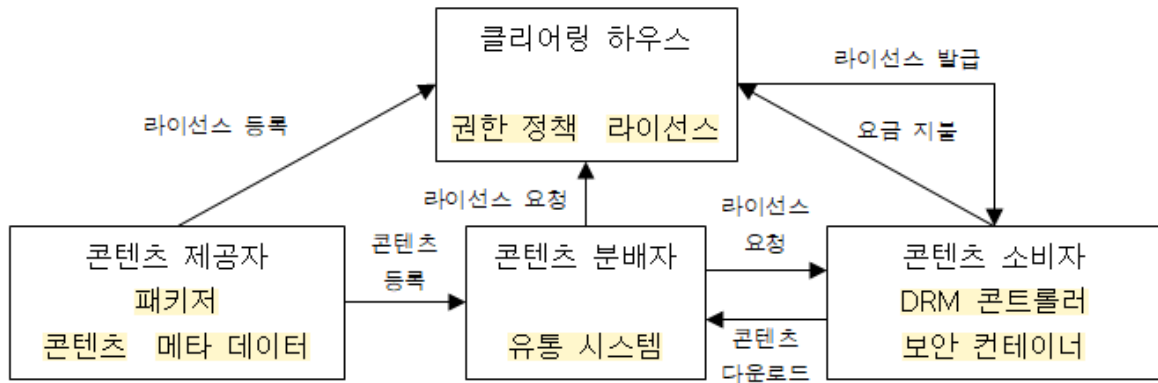
구조 패턴

- 클래스나 객체들을 조합하여 더 큰 구조로 만들 수 있게 해주는 패턴
- 어댑터 : 호환성이 없는 클래스들의 인터페이스를 다른 클래스가 이용할 수 있도록 변환해주는 패턴
- 브리지 : 구현부에서 추상층을 분리하여 서로가 독립적으로 확장할 수 있도록 구성한 패턴
- 컴포지트 : 여러 객체를 가진 복합 객체와 단일 객체를 구분 없이 다루고자 할 때 사용하는 패턴
- 데코레이터 : 객체 간의 결합을 통해 능동적으로 기능들을 확장할 수 있는 패턴
- 퍼싸드 : 복잡한 서브 클래스들을 피해 더 상위에 인터페이스를 구상함으로써 서브 클래스의 기능을 간편하게 사용할 수 있도록 하는 패턴
- 플라이웨이트 : 인스턴스가 필요할 때마다 생성하는 것이 아닌 공유해서 사용함으로써 메모리를 절약하는 패턴
- 프록시 : 접근이 어려운 객체와 여기에 연결하려는 객체 사이에서 인터페이스 역할을 수행하는 패턴

행위 패턴

- 클래스나 객체들이 서로 상호작용하는 방법이나 책임 분배 방법을 정의한 패턴
- 책임 연쇄 : 요청을 처리할 수 있는 객체가 둘 이상 존재하여 한 객체가 처리하지 못하면 다음 객체로 넘어가는 형태의 패턴
- 커맨드 : 요청을 객체의 형태로 캡슐화하여 재이용하거나 취소할 수 있도록 요청에 필요한 정보를 저장하거나 로그에 남기는 패턴
- 인터프리터 : 언어에 문법 표현을 정의하는 패턴
- 반복자 : 자료 구조와 같이 접근이 잦은 객체에 대해 동일한 인터페이스를 사용하도록 하는 패턴
- 중재자 : 수많은 객체들 간의 복잡한 상호작용을 캡슐화하여 객체로 정의하는 패턴
- 메멘토 : 특정 시점에서의 객체 내부 상태를 객체화함으로써 이후 요청에 따라 객체를 해당 시점의 상태로 돌릴 수 있는 기능을 제공하는 패턴
- 옵서버 : 한 객체의 상태가 변화하면 객체에 상속되어 있는 다른 객체들에게 변화된 상태를 전달하는 패턴
- 상태 : 객체의 상태에 따라 동일한 동작을 다르게 처리해야 할 때 사용하는 패턴
- 전략 : 동일한 계열의 알고리즘들을 개별적으로 캡슐화하여 상호 교환할 수 있게 정의하는 패턴
- 템플릿 메소드 : 상위 클래스에서 골격을 정의하고 하위 클래스에서 처리를 구체화하는 구조의 패턴
- 방문자 : 각 클래스들의 데이터 구조에서 처리 기능을 분리하여 별도의 클래스로 구성하는 패턴

디지털 저작권 관리의 흐름도



- 클리어링 하우스 : 저작권에 대한 사용 권한, 라이선스 발급, 사용량에 따른 결제 관리 등 수행
- 콘텐츠 제공자 : 콘텐츠를 제공하는 저작권자
- 패키저 : 콘텐츠를 메타 데이터와 함께 배포 가능한 형태로 묶어 암호화는 프로그램
- 콘텐츠 분배자 : 암호화된 콘텐츠를 유통
- 콘텐츠 소비자 : 콘텐츠를 구매해서 사용
- DRM 컨트롤러 : 배포된 콘텐츠의 이용 권한을 통제하는 프로그램
- 보안 컨테이너 : 콘텐츠 원본을 안전하게 유통하기 위한 전자적 보안 장치

인터페이스 구현 검증 도구

- xUnit : Java, C++, .Net 등 다양한 언어를 지원
- STAF : 서비스 호출 및 컴포넌트 재사용 등 다양한 환경을 지원
- FitNesse : 웹 기반 테스트케이스 설계, 진행, 결과 확인 등을 지원
- NTAF : FitNesse의 협업 기능과 STAF의 재사용 및 확장성을 통합한 NHN의 프레임워크
- Selenium : 다양한 브라우저 및 개발 언어 지원
- watir : Ruby를 사용

속성의 종류

- 기본 속성 : 업무 분석을 통해 정의한 속성. 가장 많고 일반적 ex) 제품명
- 설계 속성 : 업무상 존재하지 않지만 설계 과정에서 도출해내는 속성 ex) 제품 코드
- 파생 속성 : 다른 속성으로부터 계산되거나 파생된 속성 ex) 제품 판매량, 판매 수익

정규화 과정

- 1NF -> 2NF -> 3NF -> BCNF -> 4NF -> 5NF
- 1NF(제1 정규형)
 - > 릴레이션에 속한 모든 값들이 원자 값으로만 구성
- 2NF(제2 정규형)
 - > 기본키가 아닌 모든 속성이 기본키에 대하여 완전 함수적 종속을 만족
 - > 완전 함수적 종속 : 기본키에 의해서 속성이 결정
 - > 부분 함수적 종속 : 기본키의 일부에 의해 속성이 결정
 - > 아래와 같이 학번과 과목 코드가 기본키인 릴레이션이 있을 때

-> 과목 점수는 기본키(학번, 과목 코드)를 가지고 알 수 있음 = 완전 함수적 종속

-> 이름은 기본키의 일부(학번)를 가지고 알 수 있음 = 부분 함수적 종속

학번	이름	과목코드	과목점수
2015xxxx	박xx	A01	95
2017xxxx	김xx	A02	80

- 3NF(제3 정규형)

-> 기본키가 아닌 모든 속성이 기본키에 대해 이행적 종속을 만족하지 않음

-> 이행적 종속 : $A \rightarrow B, B \rightarrow C$ 일 때 $A \rightarrow C$ 를 만족하는 관계

- BCNF(Boyce-Codd정규형)

-> 결정자가 모두 후보키

- 4NF(제4 정규형)

-> 릴레이션에 다치 종속이 성립하는 경우 모든 속성이 함수적 종속 관계를 만족

- 5NF(제5 정규형)

-> 모든 조인 종속이 후보키를 통해서만 성립



트리 기반 인덱스

- 인덱스를 저장하는 블록들이 트리 구조를 이루고 있는 것으로 상용 DBMS에서는 트리 구조 기반의 B+ 트리 인덱스를 주로 활용

- B 트리 인덱스

- > 일반적으로 사용하는 인덱스 방식
- > 루트 노드에서 하위 노드로 키값의 크기를 비교하면서 데이터를 검색
- > 모든 리프 노드의 레벨은 같음
- B+ 트리 인덱스
 - > 단말 노드가 아닌 노드로 구성된 인덱스 세트와 단말 노드로만 구성된 순차 세트로 구분
 - > 인덱스 세트에 있는 노드들은 단말 노드에 있는 키 값을 찾아갈 수 있는 경로로만 제공
 - > 순차 세트에 있는 단말 노드가 해당 데이터 레코드의 주소를 가리킴
 - > 인덱스 세트에 있는 모든 키 값이 단말 노드에 다시 나타나므로 단말 노드 만을 이용한 순차 처리 가능

비트맵 인덱스

- 인덱스 칼럼의 데이터를 Bit 값인 0 또는 1로 변환하여 인덱스 키로 사용하는 방법
- 키 값을 포함하는 로우(Row)의 주소를 제공
- 데이터가 Bit로 구성되어 있어 효율적인 논리 연산이 가능하고 저장공간이 작음

함수 기반 인덱스

- 칼럼의 값 대신 칼럼의 특정 함수나 수식을 적용하여 산출된 값을 사용
- B+ 트리 인덱스 또는 비트맵 인덱스를 생성하여 사용
- 데이터를 입력하거나 수정할 때 함수를 적용하기 때문에 부하가 발생할 수 있음
- 사용자 정의 함수를 사용했을 경우 시스템 함수보다 부하가 더 크다
- 대소문자, 띄어쓰기 등에 상관없이 조회할 때 유용하게 사용

개인키 / 공개키 암호 방식

- 암호화 방식의 키와 복호화 방식의 키가 같을 때
 - > 개인키 / 비밀키 / 대칭키 암호 방식
 - > 종류 : 전위 기법, 대수 기법, 합성 기법(DES)
- 암호화 방식의 키와 복호화 방식의 키가 다를 때
 - > 공개키 / 비대칭키
 - > RSA 기법

접근통제 보안 모델

- 보안 정책을 구현하기 위한 정형화된 모델
- 기밀성 모델
 - > 군사적인 목적으로 개발된 최초의 수학적 모델로 기밀성 보장이 최우선
- 무결성 모델
 - > 기밀성 모델에서 발생하는 불법적인 정보 변경을 방지하기 위해 무결성을 기반으로 개발된 모델
 - > 데이터 일관성 유지에 중점을 두어 개발

사용자 정의 함수 실행

```
SELECT 사용자 정의 함수명 FROM 테이블명;  
INSERT INTO 테이블명(속성명) VALUES (사용자 정의 함수명);  
DELETE FROM 테이블명 WHERE 속성명 = 사용자 정의 함수명;  
UPDATE 테이블명 SET 속성명 = 사용자 정의 함수명;
```

DBMS 접속 기술

- JDBC(Java DataBase Connectivity)

- > Java 언어
- > 썬 마이크로시스템에서 출시
- > Java SE에 포함되어 있고 JDBC 클래스는 java.sql, javax.sql에 포함
- > 접속하려는 DBMS에 대한 드라이버 필요

- ODBC(Open DataBase Connectivity)

- > 개발 언어와 상관 없음
- > 마이크로소프트에서 출시
- > MS-Access, DBase, DB2, Excel, Text 등 다양한 데이터베이스에 접근 가능

- MyBatis

- > JDBC 코드를 단순화하여 사용할 수 있는 SQL Mapping 기반 오픈소스 접속 프레임워크
- > SQL 문장을 분리하여 XML 파일을 만들고 Mapping을 통해 SQL을 실행
- > SQL을 거의 그대로 사용할 수 있어 국내 환경에 적합

단문 SQL 테스트

- DDL, DML, DCL이 포함되어 있는 SQL과 TCL을 직접 실행하여 테스트

- DESCRIBE 명령어를 이용하면 DDL로 작성된 테이블이나 뷰의 속성, 자료형, 옵션들을 확인할 수 있음

- DCL로 설정된 사용자 권한은 사용자 권한 정보가 저장된 테이블을 SELECT문으로 조회하거나 SHOW 명령어로 확인할 수 있음

- > Oracle : SELECT * FROM DBA_ROLE_PRIVS WHERE GRANTEE = 사용자;
- > MySQL : SHOW GRANTS FOR 사용자@호스트;

절차형 SQL 테스트

- 프로시저, 사용자 정의 함수, 트리거 등의 절차형 SQL은 디버깅을 통해 기능의 적합성 여부 검증, 실행을 통해 결과를 확인하는 테스트를 수행

- SHOW 명령어를 통해 오류 내용을 확인

- > SHOW ERRORS;

- 데이터베이스에 변화를 줄 수 있는 SQL문은 주석 처리 후 출력문을 이용하여 결과 확인

- > Oracle : DBMS_OUTPUT.ENABLE; / DBMS_OUTPUT.PUT_LINE(데이터);
- > MySQL : SELECT 데이터;

ORM 프레임워크

- ORM을 구현하기 위한 구조와 구현을 위해 필요한 여러 기능들을 제공하는 소프트웨어
- JAVA : JPA, Hibernate, EclipseLink, DataNucleus, Ebean 등
- C++ : ODB, QxOrm 등
- Python : Django, SQLAlchemy, Storm 등
- iOS : DatabaseObjects, Core Data 등
- .NET : NHibernate, DatabaseObjects, Dapper 등
- PHP : Doctrine, Propel, RedBean 등

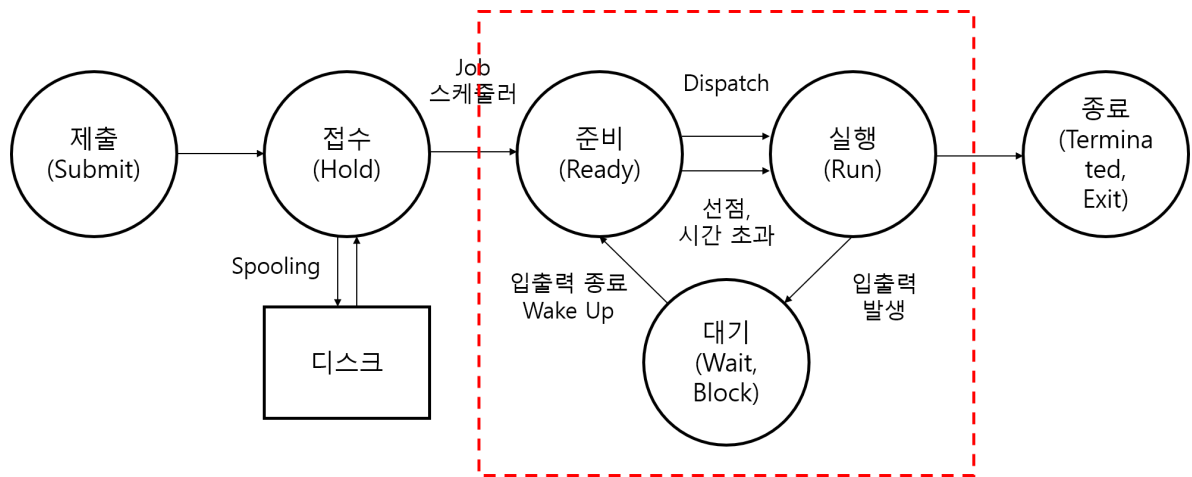
배치 스케줄러

- 일괄 처리 작업이 설정된 주기에 맞춰 자동으로 수행되도록 지원해주는 도구
- 잡 스케줄러라고도 함
- 스프링 배치
 - > Spring Source사와 Accenture사가 공동 개발한 오픈소스 프레임워크
 - > 스프링 프레임워크의 특성을 그대로 가져와 스프링의 기능을 모두 사용할 수 있음
 - > 데이터베이스나 파일의 데이터를 교환하는데 필요한 컴포넌트를 제공
 - > 로그 관리, 추적, 트랜잭션 관리, 작업 처리 통계, 작업 재시작 등의 다양한 기능 제공
 - > 구성요소 : Job, Job Launcher, Step, Job Repository
- Quartz
 - > 스프링 프레임워크로 개발되는 응용 프로그램들의 일괄 처리를 위한 다양한 기능을 제공하는 오픈소스 라이브러리
 - > 수행할 작업과 수행 시간을 관리하는 요소들을 분리하여 일괄 처리 작업에 유연성을 제공
 - > 구성요소 : Scheduler, Job, JobDetail, Trigger

PCB(Process Control Block)

- 운영체제가 프로세스에 대한 중요한 정보를 저장해 놓은 곳
- 프로세스가 생성될 때마다 고유의 PCB를 생성하고 프로세스 완료 시 제거됨
- PCB에 저장되는 정보
 - > 프로세스 현재 상태
 - > 포인터
 - > 프로세스 고유 식별자
 - > 스케줄링 및 프로세스의 우선순위
 - > CPU 레지스터 정보
 - > 주기억장치 관리 정보
 - > 입출력 상태 정보
 - > 계정 정보

프로세스 상태 전이



- 준비

- > 프로세스가 프로세서를 할당받기 위해 기다리고 있는 상태
- > 프로세스는 준비상태 큐에서 실행 준비
- > Job 스케줄러에 의해 수행

- 실행

- > 준비상태 큐에 있는 프로세스가 프로세서를 할당받아 실행되는 상태
- > 프로세스 수행이 완료되기 전에 프로세스에게 주어진 할당 시간이 종료되면 프로세스는 준비 상태로 전이
- > 실행 중인 프로세스에 입출력 처리가 필요하면 실행 중인 프로세스는 대기상태로 전이
- > CPU 프로세스에 의해 수행

- 대기 : 프로세스에 입출력 처리가 필요하면 현재 실행 중인 프로세스가 중단되고, 입출력 처리가 완료될 때까지 대기하고 있는 상태

프로세스 상태 전이 관련 용어

- Dispatch : 준비 상태에 대기하고 있는 프로세스 중 하나가 프로세스를 할당받아 실행 상태로 전이되는 과정
- Wake up : 입출력 작업이 완료되어 프로세스가 대기 상태에서 준비 상태로 전이되는 과정
- Spooling : 입출력 장치의 공유 및 상대적으로 느린 입출력장치의 처리 속도를 보완하기 위해 입출력할 데이터를 직접 장치에 보내지 않고 나중에 한꺼번에 입출력하기 위해 디스크에 저장하는 과정

스케줄링의 개요

- 문맥 교환

- > 하나의 프로세스에서 다른 프로세스로 CPU가 할당되는 과정에서 발생하는 것

프로세스 스케줄링의 기법

- 비선점 스케줄링

- > 강제 종료 불가능
- > 이미 할당된 CPU를 다른 프로세스가 강제로 빼앗아 사용할 수 없음
- > 프로세스가 CPU를 할당받으면 완료될 때 까지 사용
- > 프로세스 응답 시간 예측이 용이
- > 일괄 처리 방식에 적합

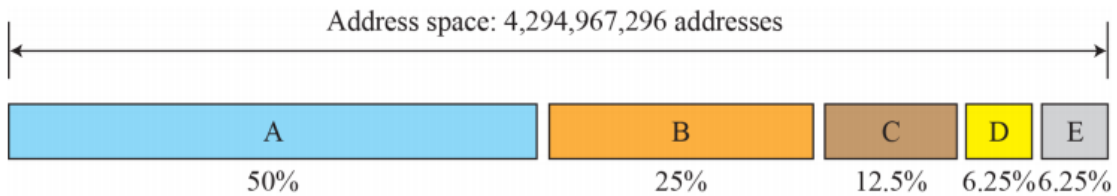
-> 중요한(짧은) 작업이 중요하지 않은(긴) 작업을 기다리는 경우가 발생할 수 있음

-> FCFS, SJF, 우선순위, HRN, 기한부 등의 알고리즘

IP주소(Internet Protocol Address)

- 인터넷에 연결된 컴퓨터를 구분하기 위한 고유한 주소

- 8비트 씩 4부분으로 구성됨(IPv4)



8 bits

8 bits

8 bits

8 bits

Class A

0 Prefix

Suffix

Class B

10 Prefix

Suffix

Class C

110 Prefix

Suffix

Class D

1110 Multicast addresses

Class E

1111 Reserved for future use

Class	Prefixes	First byte
A	$n = 8$ bits	0 to 127
B	$n = 16$ bits	128 to 191
C	$n = 24$ bits	192 to 223
D	Not applicable	224 to 239
E	Not applicable	240 to 255

OSI 모델의 계층

- 물리 계층

-> 전송에 필요한 두 장치 간의 실제 접속과 절단 등 기계적, 전기적, 기능적, 절차적 특성에 대한 규칙을 정의

-> 물리적 전송 매체와 신호 방식을 정의

-> RS-232C, X.21 등의 표준

-> 관련 장비 : 리피터, 허브

-> 데이터 단위 : 비트

- 데이터 링크 계층

-> 두 개의 인접 개방 시스템들 간의 신뢰성 있고 효율적인 정보 전송을 할 수 있도록 함

-> 주요 기능 : 흐름 제어, 프레임 동기화, 오류 제어, 순서 제어

-> HDLC, LAPB, LLC, MAC, LAPD, PPP 등의 표준

-> 관련 장비 : 브리지

-> 데이터 단위 : 프레임

- 네트워크 계층

-> 개방 시스템들 간의 네트워크 연결을 관리하고 데이터의 교환 및 중계 기능

-> 네트워크 연결 설정, 유지, 해제

-> 경로 설정, 데이터 교환 및 중계, 트래픽 제어, 패킷 정보 전송 수행

-> X.25, IP 등의 표준

-> 관련 장비 : 라우터

-> 데이터 단위 : 패킷

- 전송 계층

-> 논리적 안정과 균일한 데이터 전송 서비스를 제공

-> 종단 시스템 간의 투명한 데이터 전송을 가능하게 함

-> 종단 시스템 간의 전송 연결 설정, 데이터 전송, 연결 해제 기능

-> 주소 설정, 다중화, 오류 제어, 흐름 제어

-> TCP, UDP 등의 표준

-> 관련 장비 : 게이트웨이

-> 데이터 단위 : 세그먼트

- 세션 계층

-> 송수신 측 간의 관련성을 유지하고 대화 제어를 담당

-> 동기점 : 송수신 측간의 대화 동기를 위해 전송하는 정보를 일정한 부분에 두어 정보의 수신 상태를 체크하는 포인트

-> 데이터 단위 : 메시지

- 표현 계층

-> 데이터를 응용 계층, 세션 계층에 보내기 전에 계층에 맞게 변환

-> 서로 다른 데이터 표현 형태를 갖는 시스템 간 상호 접속을 위해 필요한 계층

-> 코드 변환, 데이터 암호화, 데이터 압축, 구문 검색, 정보 형식 변환, 문맥 관리 기능

-> 데이터 단위 : 메시지

- 응용 계층

-> 사용자가 OSI 환경에 접속할 수 있도록 서비스 제공

-> 프로세스 간의 정보 교환, 전자 사서함, 가상 터미널 등의 서비스 제공

구조적 방법론

- 정형화된 분석 절차에 따라 사용자 요구사항을 파악하여 문서화하는 처리 중심의 방법론

정보공학 방법론

- 정보 시스템 개발을 위해 계획, 분석, 설계, 구축에 정형화된 기법들을 상호 연관성 있게 통합 및 적용하는 자료 중심의 방법론

객체지향 방법론

- 기계의 부품을 조립하듯이 객체들을 조립하여 소프트웨어를 구현하는 방법론

컴포넌트 기반(CBD) 방법론

- 컴포넌트를 조합하여 새로운 애플리케이션을 만드는 방법론

애자일 방법론

- 고객의 요구사항 변화에 빠르고 유연하게 대응하도록 일정한 주기를 반복하면서 개발 과정을 진행하는 방법론

제품 계열 방법론

- 특정 제품에 적용하고 싶은 공통된 기능을 정의하여 개발하는 방법론
- 임베디드 소프트웨어 개발에 적합

소프트웨어 비용 결정 요소

- 프로젝트 요소 : 제품 복잡도, 시스템 크기, 요구되는 신뢰도
- 자원 요소 : 인적 자원, 하드웨어 자원, 소프트웨어 자원
- 생산성 요소 : 개발자 능력, 개발 기간

소프트웨어 개발 표준의 개요

- 소프트웨어 개발 단계에서 수행하는 품질 관리에 사용되는 국제 표준

ISO/IEC 12207

- ISO에서 만든 표준 소프트웨어 생명 주기 프로세스
- 소프트웨어의 개발, 운영, 유지보수를 관리하기 위한 생명 주기 표준을 제공
- 기본 / 생명 / 조직 생명 주기 프로세스로 구분

CMMI(Capability Maturity Model Integration, 능력 성숙도 통합 모델)

- 소프트웨어 개발 조직의 업무 능력 및 조직의 성숙도를 평가
- 성숙도는 초기, 관리, 정의, 정량적 관리, 최적화로 구분

SPICE(Software Process Improvement and Capability dEtermination, 소프트웨어 처리 개선 및 능력 평가 기준)

- 소프트웨어의 품질 및 생산성 향상을 위해 소프트웨어 프로세스를 평가 및 개선하는 국제 표준
- ISO/IEC 15504

소프트웨어 개발 방법론 테일러링 고려사항

- 내부적 요건 : 목표 환경, 요구사항, 프로젝트 규모, 보유 기술
- 외부적 요건 : 법적 제약사항, 표준 품질 기준

신기술 관련 용어

네트워크 관련 신기술 용어

- 인터클라우드 컴퓨팅 : 각기 다른 클라우드 서비스를 연동하거나 컴퓨팅 자원의 동적 할당이 가능하도록 여러 클라우드 서비스 제공자들이 제공하는 클라우드 서비스나 자원을 연결하는 기술
- 매시 네트워크 : 차세대 이동통신, 홈네트워킹, 공공 안전 등 특수 목적을 위한 새로운 방식의 네트워크 기술로 대규모 디바이스의 네트워크 생성에 최적화
- NDN(Named Data Networking) : 콘텐츠 자체의 정보와 라우터 기능만으로 데이터 전송을 수행하는 기술
- NGN(Next Generation Network, 차세대 통신망) : 유선망 기반의 차세대 통신망으로 유선망뿐만 아니라 이동 사용자를 목표로 하며 이동 통신에서 제공하는 완전한 이동성 제공을 목표로 개발

- SDN(Software Defined Networking) : 네트워크를 컴퓨터처럼 모델링하여 여러 사용자가 각각의 소프트웨어들로 네트워킹을 가상화하여 제어하고 관리하는 네트워크
- UWB(Ultra WideBand) : 짧은 거리에서 많은 양의 디지털 데이터를 낮은 전력으로 전송하기 위한 무선 기술로 무선 디지털 펄스라고도 함
- 피코넷 : 여러 개의 독립된 통신 장치가 블루투스 기술이나 UWB 통신 기술을 사용하여 통신망을 형성하는 무선 네트워크 기술
- GIS(Geographic Information System) : 지리적인 자료를 수집, 저장, 분석, 출력할 수 있는 컴퓨터 응용 시스템으로 위성을 이용해 모든 사물의 위치 정보를 제공해줌
- USN(Ubiquitous Sensor Network) : 각종 센서로 수집한 정보를 무선으로 수집할 수 있도록 구성하는 네트워크
- SON(Self Organizing Network) : 주변 상황에 맞추어 스스로 망을 구성하는 네트워크
- 애드 혹 네트워크 : 재난 현장과 같이 별도의 고정된 유선망을 구축할 수 없는 장소에서 모바일 호스트만을 이용하여 구성한 네트워크
- 네트워크 슬라이싱 : 여러 글로벌 이동통신 표준화 단체가 선정한 5G의 핵심기술 중 하나로 네트워크에서 하나의 물리적인 코어 네트워크 인프라를 독립된 다수의 가상 네트워크로 분리하여 각각의 네트워크를 통해 다양한 고객 맞춤형 서비스를 제공하는 것을 목적으로 하는 네트워크 기술
- 지능형 초연결망 : 스마트 시티, 스마트 스테이션 등 4차 산업혁명 시대를 맞아 새로운 변화에 따라 급격하게 증가하는 데이터 트래픽을 효과적으로 수용하기 위해 시행되는 정부 주관 사업

SW 관련 신기술 용어

- 분산 원장 기술 : 중앙 관리자나 중앙 데이터 저장소가 존재하지 않고 P2P 망내의 참여자들에게 모든 거래 목록이 분산 저장되어 거래가 발생할 때마다 지속적으로 갱신되는 디지털 원장
- 양자 암호키 분배 : 양자 통신을 위해 비밀키를 분배하여 관리하는 기술
- 프라이버시 강화 기술 : 개인 정보 위험 관리 기술
- 공통 평가 기준 : ISO 15408 표준으로 채택된 정보 보호 제품 평가 기준
- 개인정보 영향평가 제도 : 개인정보를 활용하는 새로운 정보 시스템의 도입 및 기존 정보시스템의 중요한 변경 시 시스템의 구축, 운영이 기업의 고객은 물론 국민의 사생활에 미칠 영향에 대해 미리 조사, 분석, 평가하는 제도
- 그레이웨어 : 소프트웨어를 제공하는 입장에서는 악의적이지 않은 유용한 소프트웨어 일지라도 사용자 입장에서는 유용할 수도 있고 악의적일 수도 있는 악성 코드나 공유 웨어
- 매시업 : 웹에서 제공하는 정보 및 서비스를 이용하여 새로운 소프트웨어나 서비스 데이터베이스 등을 만드는 기술
- 시맨틱 웹 : 컴퓨터가 사람을 대신하여 정보를 읽고 이해하고 가공하여 새로운 정보를 만들어 낼 수 있도록 이해하기 쉬운 의미를 가진 차세대 지능형 웹
- 오픈 그리드 서비스 아키텍처 : 애플리케이션 공유를 위한 웹 서비스를 그리드 상에서 제공하기 위해 만든 개방형 표준
- 서비스 지향 아키텍처 : 기업의 소프트웨어 인프라인 정보시스템을 공유와 재사용이 가능한 서비스 단위나 컴포넌트 중심으로 구축하는 정보기술 아키텍처
- 소프트웨어 에스크로 : 소프트웨어 개발자의 지적재산권을 보호하고 사용자는 저렴한 비용으로 소프트웨어를 안정적으로 사용 및 유지보수받을 수 있도록 소스 프로그램과 기술 정보 등을 제3의 기관에 보관하는 것

- 복잡 이벤트 처리 : 실시간으로 발생하는 많은 사건들 중 의미가 있는 것만을 추출할 수 있도록 사건 발생 조건을 정의하는 데이터 처리 방법

- 디지털 트윈 : 현실속의 사물을 소프트웨어로 가상화한 모델

HW 관련 신기술 용어

- 고가용성 : 긴 시간동안 안정적인 서비스 운영을 위해 장애 즉시 다른 시스템으로 대체 가능한 환경을 구축하는 메커니즘

- RAID : 여러 개의 하드디스크로 디스크 배열을 구성하여 파일을 구성하여 파일을 구성하고 있는 데이터 블록들을 서로 다른 디스크들에 분산 저장하는 기술

- 컴패니언 스크린 : TV 방송 시청 시 방송 내용을 공유하며 추가적인 기능을 수행할 수 있는 디바이스

- 멤스 : 초정밀 반도체 기술을 바탕으로 센서나 액추에이터 등 기계 구조를 다양한 기술로 미세 가공하여 전기기계적 동작을 할 수 있도록 한 초미세 장치

- 트러스트존 기술 : ARM에서 개발한 하나의 프로세서 내에 일반 애플리케이션을 처리하는 일반 구역과 보안이 필요한 애플리케이션을 처리하는 보안 구역으로 분할하여 관리하는 하드웨어 기반의 보안 기술

- 엠디스크 : 한 번의 기록만으로도 자료를 영구 보관할 수 있는 광 저장 장치

- 멤리스터 : 메모리와 레지스터의 합성어로 전류의 방향과 양 등 기존의 경험을 모두 기억하는 소자

DB 관련 신기술 용어

- 브로드 데이터 : 다양한 채널에서 소비자와 상호 작용을 통해 생산 되었거나 기업 마케팅에 있어 효율적이고 다양한 데이터, 이전에 사용하지 않거나 몰랐던 새로운 데이터나 기존 데이터에 새로운 가치가 더해진 데이터

- 메타 데이터 : 일련의 데이터를 정의하고 설명해주는 데이터

- 디지털 아카이빙 : 디지털 정보 자원을 장기적으로 보존하기 위한 작업

- 하둡 : 오픈 소스를 기반으로 한 분산 컴퓨팅 플랫폼

- 타조 : 우리나라가 주도적으로 개발 중인 하둡 기반의 분산 데이터 웨어하우스 프로젝트

- 데이터 다이어트 : 데이터를 삭제하는 것이 아닌 압축하고 중복된 정보를 배제하고 새로운 기분에 따라 나누어 저장하는 작업

경로 제어 프로토콜

- 효율적인 경로 제어를 위해 네트워크를 제어하는 프로토콜

- IGP(내부 게이트웨이 프로토콜)

-> 하나의 자율 시스템(AS) 내의 라우팅에 사용

-> RIP : 현재 가장 널리 사용되며 소규모 네트워크에서 효율적인 방법

-> OSPF : 대규모 네트워크에서 많이 사용되는 프로토콜

- EGP(외부 게이트웨이 프로토콜)

-> 자율 시스템 / 게이트웨이 간 라우팅에 사용되는 프로토콜

- BGP

-> 자율 시스템 간의 라우팅 프로토콜

-> EGP의 단점을 보완하기 위해 만들어짐

-> 초기 연결 시 라우팅 테이블을 교환하고 이후에는 변화된 정보만을 교환

흐름 제어

- 네트워크 내의 원활한 흐름을 위해 송수신 사이 전송되는 패킷의 양이나 속도를 규제

- 송수신간 처리 속도 또는 버퍼 크기의 차이에 의해 생길 수 있는 버퍼 오버플로우를 방지

- 정지-대기 : 수신 측에서 확인 신호를 받아야 다음 패킷을 전송

- 슬라이딩 윈도우

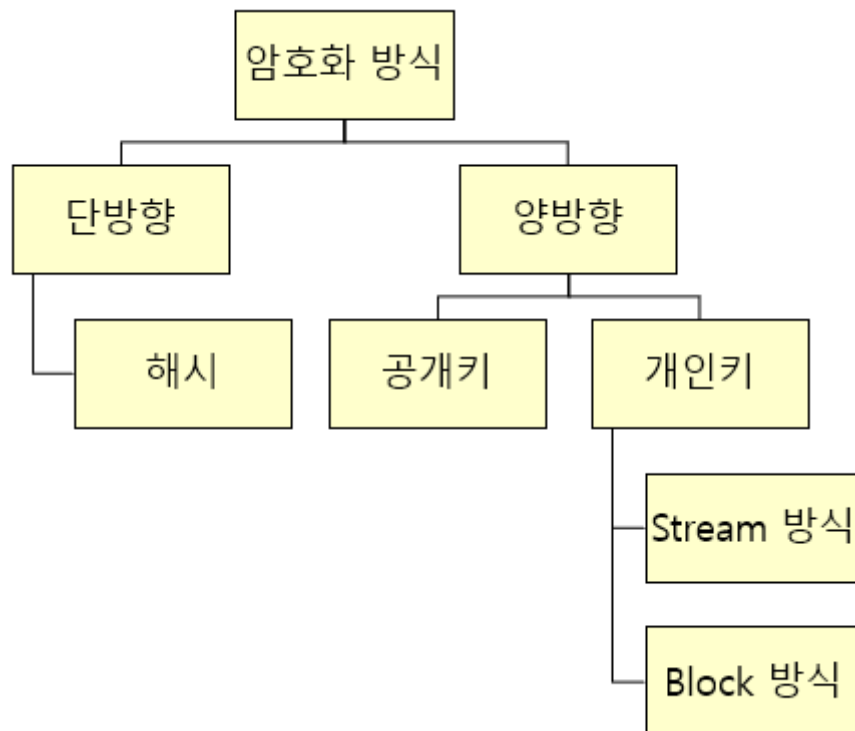
-> 확인 신호를 이용하여 송신 데이터의 양을 조절

-> 수신 측의 확인 신호를 받지 않아도 정해진 패킷 수만큼 연속적으로 전송하는 방식

-> 한 번에 여러 개의 패킷을 전송할 수 있음

암호 알고리즘의 개요

- 중요정보를 보호하기 위한 평문을 암호화된 문장으로 만드는 방법



양방향 암호화 알고리즘 종류

- SEED : 한국인터넷진흥원, 블록 크기는 128비트, 키의 길이에 따라 128, 256로 분류

- ARIA : 국가정보원, 블록 크기는 128비트, 키의 길이에 따라 128, 192, 256로 분류

- DES : 미국 NBS, 블록 크기는 64비트, 키의 길이 56비트

- AES : 미국 표준 기술 연구소, 블록 크기는 128비트, 키의 길이에 따라 128, 192, 256로 분류

서비스 거부 공격의 개념

- 표적이 되는 서버의 자원을 고갈시킬 목적으로 다수의 공격자 또는 시스템에서 대량의 데이터를 한 곳의 서버를 집중적으로 전송함으로써 표적이 되는 서버의 정상적인 기능을 방해

Ping of Death

- Ping 명령 전송 시 패킷의 크기를 인터넷 프로토콜 허용 범위 이상으로 공격하여 공격 대상의 네트워크를 마비시키는 서비스 거부 방법

SMURFING

- IP나 ICMP의 특성을 악용하여 엄청난 양의 데이터를 한 사이트에 집중적으로 보냄으로써 네트워크를 불능 상태로 만드는 공격 방법

SYN Flooding

- 공격자가 가상의 클라이언트로 위장하여 3-way-handshake 과정을 의도적으로 중단시킴으로써 공격 대상지인 서버가 대기 상태에 놓여 정상적인 서비스를 수행하지 못하게 하는 공격 방법

TearDrop

- 데이터의 송수신 단계에서 전송되는 Fragment Offset 값을 변경시켜 패킷을 재조립할 때 오류로 인한 과부하를 발생시킴으로 시스템이 다운되도록 하는 공격 방법

Land

- 패킷 전송 시 송수신 IP 주소를 모두 공격 대상의 IP주소로 하여 공격 대상에게 전송하여 무한히 자신에게 응답을 수행하게 되는 공격 방법

DDos(Distributed Denial of Service, 분산 서비스 거부) 공격

- 여러 곳에 분산된 공격 지점에서 한 곳의 서버에 대해 공격을 수행

- 네트워크에서 취약점이 있는 호스트들을 탐색한 후 호스트들에게 분산 서비스 공격 툴을 설치하여 에이전트로 만든 후 공격에 이용

- 분산 서비스 공격 툴

-> Trin00 : 초기 형태의 데몬으로 UDP Flooding 공격 수행

-> TFN : UDP Flooding, TCP SYN Flood 공격, ICMP 응답 요청, 스머핑 공격 등 수행

-> TFN2K : TFN의 확장판

-> Stacheldraht : 이전의 툴들을 유지하면서 암호화된 통신을 수행하며 툴이 자동으로 업데이트되도록 설계

네트워크 침해 공격 관련 용어

- 스미싱 : 문자 메시지를 이용해 사용자의 개인 신용 정보를 빼내는 수법

- 스피어 피싱 : 일반적인 이메일로 위장한 메일을 지속적으로 발송하여 메일의 링크나 첨부된 파일을 클릭하게 유도하여 개인 정보를 탈취

- APT(지능형 지속 위협) : 조직적으로 특정 기업이나 조직 네트워크에 침투해 활동 거점을 마련한 뒤 때를 기다리면서 보안을 무력화시키고 정보를 수집한 다음 외부로 빼돌리는 형태의 공격

- 무작위 대입 공격 : 암호화된 문서의 암호키를 찾기 위해 무작위로 값을 대입하여 공격하는 방식

- 쿼싱 : QR코드를 통해 악성 앱을 다운받게 하여 개인 정보를 탈취하는 공격 방식
- SQL 삽입 공격 : 웹사이트를 무차별적으로 공격하는 과정에서 취약한 사이트 발견 시 데이터를 조작하는 일련의 공격 방식
- 크로스 사이트 스크립 : 웹 페이지의 내용을 사용자 브라우저에 표현하기 위해 사용되는 스크립트의 취약점을 악용한 해킹 기법

정보 보안 침해 공격 관련 용어

- 좀비 PC : 악성코드에 감염되어 다른 프로그램이나 컴퓨터를 조종하도록 만들어진 컴퓨터
- C&C 서버 : 해커가 원격지에서 감염된 좀비 PC에 명령을 내리고 악성코드를 제어하기 위한 용도로 사용하는 서버
- 봇넷 : 악성 프로그램에 감염된 컴퓨터들이 네트워크로 연결된 형태
- 웜 : 네트워크를 통해 연속적으로 자신을 복사하여 시스템의 부하를 높여 시스템을 다운시키는 바이러스의 일종
- 제로 데이 공격 : 보안 취약점이 발견됐을 때 공표되기도 전에 해당 취약점을 통해 신속하게 이루어지는 보안 공격
- 키로거 공격 : 사용자의 키보드 움직임을 탐지하여 개인 정보를 몰래 빼가는 공격
- 랜섬웨어 : 사용자의 컴퓨터에 잡입해 파일을 암호화하여 사용자가 열지 못하게 하는 프로그램
- 백도어 : 액세스 편의를 위해 시스템 보안을 제거하여 만들어 놓은 비밀 통로를 통해 범죄에 악용되는 형태
- 트로이 목마 : 정상적인 기능을 하는 프로그램인 척 프로그램에 숨어 있다가 해당 프로그램이 동작될 때 활성화되어 부작용을 일으키는 형태

보안 솔루션의 개념

- 접근 통제, 침입 차단 등을 수행하여 외부로부터 불법적인 침입을 막는 기술 및 시스템

방화벽

- 기업이나 조직 내부의 네트워크와 인터넷 간에 전송되는 정보를 선별하여 수용, 거부, 수정하는 기능을 가진 침입 차단 시스템

침입 탐지 시스템(IDS)

- 컴퓨터 시스템의 비정상적인 행위를 실시간으로 탐지하는 시스템
- 문제 발생 시 모든 내외부 정보의 흐름을 실시간으로 차단하기 위해 해커 침입 패턴에 대한 추적과 유해 정보 감시가 필요

침입 방지 시스템(IPS)

- 방화벽과 침입 탐지 시스템을 결합
- 비정상적인 트래픽을 능동적으로 차단하고 격리하는 방어 조치를 취하는 보안 솔루션

데이터 유출 방지(DLP)

- 내부 정보의 외우 유출을 방지하는 보안 솔루션
- 내부 PC와 네트워크 상의 모든 정보를 검색하고 사용자 행위를 탐지, 통제해 외부로의 유출을 사전에 방지

웹 방화벽

- 일반 방화벽이 탐지하지 못하는 SQL 삽입 공격, XSS 등의 웹 기반 공격을 방어할 목적으로 만들어진 웹 서버에 특화된 방화벽

VPN(가상 사설 통신망)

- 인터넷 등 통신 사업자의 공중 네트워크와 암호화 기술을 이용하여 사용자가 마치 자신의 전용 회선을 사용하는 것처럼 해주는 보안 솔루션

NAC

- 네트워크에 접속하는 내부 PC의 MAC 주소를 IP 관리 시스템에 등록 후 일관된 보안 관리 기능을 제공하는 보안 솔루션

ESM

- 다양한 장비에서 발생하는 로그 및 보안 이벤트를 통합하여 관리하는 보안 솔루션