

<<IP 주소 클래스>>

- * A Class: 국가나 대형 통신망에 사용. 0~127로 시작. 2^{24} 개의 호스트 사용 가능
- * B Class: 중대형 통신망에 사용. 128~191로 시작. 2^{16} 개의 호스트 사용 가능
- * C Class: 소규모 통신망에 이용. 192~223로 시작. 2^8 개의 호스트 사용 가능
- * D Class: 멀티캐스트용. 224~239로 시작
- * E Class: 실험적 주소이며 공용되지 않음

<<데이터모델 구성요소>> - 연구제

1과목

<<객체지향 SOLID 원칙>>

SRP; 단일 책임 원칙 Single Responsibility Principle	SW 컴포넌트는 단 하나의 책임만을 가져야 한다.
OCP; 개방 폐쇄 원칙 Open-Close Principle	확장에 대해서는 열려 있어야 하고, 수정에 대해서는 닫혀 있어야 한다.
LSP; 리스코프 치환 원칙 Liskov Substitution Principle	자식클래스는 부모클래스에서 가능한 행위를 수행할 수 있어야 한다.
ISP; 인터페이스 분리 원칙 Interface Segregation Principle	하나의 일반적인 인터페이스보단 여러 개의 구체적인 인터페이스가 낫다.
DIP; 의존관계 역전 원칙 Dependency Inversion Principle	의존 관계를 맺을 때, 변화하기 어려운 것에 의존해야 한다.

<<럼바우 모델링>> - 객동기

Object Modeling	Information Modeling. 시스템에서 요구하는 객체를 찾고 객체 간의 관계를 정의하여 ER 다이어그램을 만드는 과정까지의 모델링 가장 중요하며 선행되어 진행 객체 다이어그램 을 활용하여 표현
Dynamic Modeling	시간의 흐름 에 따라 객체들 사이의 제어 흐름, 동작 순서 등의 동적 행위를 표현 상태 다이어그램 을 활용하여 표현
Functional Modeling	프로세스들의 자료 흐름을 중심으로 처리 과정을 표현하는 모델링 자료 흐름도(DFD) 를 활용하여 표현

<<애자일 방법론 유형>> - 사람이 중심이 되어 변화에 유연하고 신속하게 적응 ↔ 폭포수

XP	의사소통 개선, 즉각적 피드백 으로 소프트웨어 품질을 높임
SCRUM	매일 정해진 시간, 장소 에서 짧은 시간 개발 을 하는 팀을 위한 프로젝트 관리
LEAN	도요타 린 시스템 품질기법을 개발 프로세스에 적용해 낭비 요소 제거

XP의 5가지 가치: 용기 단순성 의사소통 피드백 존중

▼ XP의 12가지 기본원리:

작 프로그래밍	개발자 둘이서 작으로 코딩하는 원리
지속적인 통합(CI)	매일 여러 번씩 소프트웨어를 통합하고 빌드해야 한다는 원리
메타포어(Metaphor)	공통적인 이름 체계와 시스템 서술서를 통해 고객과 개발자 간의 의사소통을 원활하게 한다는 원리
테스트 기반 개발(TDD)	작성할 프로그램에 대한 테스트를 먼저 수행하고, 이 테스트를 통과할 수 있도록 실제 프로그램 코드를 작성한다는 원리
리팩토링	프로그램의 기능을 바꾸지 않으면서 중복 제거, 단순화 등을 위해 시스템을 재구성한다는 원리

<<객체 지향 구성요소>> - 클객 메 메인속

<<객체 지향 기법>> - 캡상다추정관

<<비용산정 모형 종류>>

- ▼ 하향식: 경험 많은 전문가에게 비용 산정 의뢰 또는 여러 전문가와 조정자를 통합
 - 전문가 판단, 델파이 기법
- ▼ 상향식: 세부적인 요구사항, 기능에 따라 필요한 비용 계산

LoC	$(o+4m+p)/6$
Man Month	Man Month = LoC / 월간 생산성 기간 = Man Month / 인력
COCOMO	Organic Mode: 5만 라인 이하 Semi-Detached Mode: 30만 라인 이하 Embedded Mode: 30만 라인 이상
Putnam	SW 개발주기 단계별로 요구 인력의 분포를 가정함 (Rayleigh-Norden)
FP	$FP = \text{총 기능점수} * \{0.65 + (0.1 * \text{총 영향도})\}$

<<소프트웨어 아키텍처 4+1 뷰>> - 유논프구배

유스케이스	아키텍처 도출 및 설계, 다른 뷰 검증 : 사용자,설계자,개발자,테스트 관점
논리	기능적 요구사항, 클래스 다이어그램 : 설계자,개발자 관점
프로세스	비기능적 요구사항, 상호작용 등 관계를 표현 : 개발자,시스템통합자 관점
구현	정적 모듈 작용, 컴포넌트 다이어그램
배포	컴포넌트가 물리적 아키텍처에 어떻게 배치되는지 매핑. 배포 다이어그램

<<소프트웨어 아키텍처 패턴 유형>> Layered, Client-Server, Pipe-Filter, Broker, MVC

<<객체지향 디자인 패턴>>

- * 생성: 객체 인스턴스 생성에 관여, 클래스 정의와 객체 생성 방식 구조화, 캡슐화 수행
- * 구조: 더 큰 구조 형성 목적으로 클래스나 객체의 조합을 다루는 패턴
- * 행위: 클래스나 객체들이 상호작용하는 방법과 역할 분담을 다루는 패턴
- ▼ 생성 - 생빌 프로 팩앱싱

Builder	복잡한 인스턴스를 조립하여 만드는 구조 복합 객체를 생성할 때 객체의 생성 방법과 구현 방법을 분리하여 동일한 생성 절차에서 다른 표현 결과를 만듦 (객체의 생성과 구현을 분리해 복잡한 객체 생성)
Prototype	처음부터 일반적인 원형을 만들어 놓고, 그것을 복사한 후 필요한 부분만 수정해서 사용하는 패턴 (기존 객체를 복제하여 객체 생성)
Factory method	상위 클래스에서 객체를 생성하는 인터페이스를 정의 하위 클래스에서 인스턴스 생성 (생성할 객체의 클래스에 국한되지 않고 객체 생성)
Abstract Factory	구체적인 클래스에 의존하지 않고 서로 연관되거나 의존적인 객체들 조합을 만드는 인터페이스를 제공하는 패턴
Singleton	전역 변수를 사용하지 않고 객체를 하나만 사용하도록 하며, 생성된

	객체를 어디서든 참조할 수 있음(한 클래스에 한 객체만 존재)
--	------------------------------------

▼ 구조 - 구브데 퍼플 프록 컴 어

Bridge	기능과 구현 간 클래스 계층 연결. 추상/실제 구현 부분 독립적으로 확장 (구현 뿐 만 아니라 추상화 부분까지 변경해야 하는 경우 활용)
Decorator	기존 클래스에 필요한 기능 추가. 상속의 대안 (객체 결합을 통해 기능을 동적으로 유연하게 확장)
Facade	복잡한 시스템에 단순 인터페이스 제공으로 결합도를 낮춰 구조파악이 쉬움 오류 단위별로 확인 가능. 단순한 인터페이스 제공으로 접근성 높임
Flyweight	다수 객체의 공통 요소를 클래스화로 공유해 메모리 절약. 클래스 경량화 (여러 개의 가상 인스턴스를 제공해 메모리 절감)
Proxy	대리 객체 로 실제 객체에 접근하기 이전에 필요한 행동을 취할 수 있음. 미리 할당하지 않아도 상관 없는 것들을 실제 이용 시 할당해 메모리 절약 및 정보 은닉 역할 수행 (특정 객체로의 접근을 제어)
Composite	객체 간 관계를 트리 구조 로 구성. 부분-전체 계층 표현 (복합 객체와 단일 객체를 동일하게 취급)
Adapter	기존 클래스를 재사용하도록 중간 역할 인터페이스 생성 상속을 이용하는 클래스 패턴과 위임을 이용하는 인스턴스 패턴을 사용 (미호환 클래스들을 이용하기 위해 타 클래스의 인터페이스를 기존 인터페이스에 덧씌움)

▼ 행위 - 행 미인이 템옵 스테 비커 스트 메체

Mediator	객체의 수가 많아져 복잡해짐을 방지하기 위해 중재자 를 두어 객체 지향 목표 달성 (상호작용의 유연한 변경을 지원)
Interpreter	언어의 다양한 해석 , 구문을 나눔, 문법 자체를 캡슐화해 사용 (여러 형태의 언어 구문을 해석할 수 있음)
Iterator	컬렉션 구현 방법 및 내부 구조 유출 없이 모든 항목에 접근 (내부를 노출하지 않고 복잡한 객체의 원소에 순차적으로 접근)
Template method	작업 처리의 일부분을 서브 클래스로 캡슐화해 전체 구조는 바꾸지 않으면서 특정 단계에서 수행하는 내역을 바꿈 상위(추상)클래스는 기능 골격 제공, 하위(구체)클래스는 처리를 구체화. 코드의 양을 줄이고 유지보수가 용이 해짐 (상위 작업의 구조 변경 없이 서브클래스 로 작업 일부분 수행)
Observer	객체에 의존하는 다른 객체들에게 연락/자동내용 갱신/느슨한 결합/ 일대다 의존
State	객체 상태 캡슐화하여 클래스화 / 행위 내용 변경 원시 코드 수정 최소화/유지보수 편의성 (객체의 상태에 따라 행위 내용을 변경)
Visitor	처리 기능 분리 / 메서드가 클래스를 돌아다니며 작업을 수행 (구조변경 없이 기능만 추가/확장 할 때 사용)
Command	실행될 기능을 캡슐화하여 재사용성이 높은 클래스를 설계함. 추상클래스에 메서드를 만들어 명령에 맞는 서브클래스 실행 (요구사항 을 객체로 캡슐화)

Strategy	알고리즘 군 정의/ 알고리즘을 캡슐화 해 필요 시 교환해서 사용 (행위 객체를 클래스로 캡슐화해 동적으로 행위를 변환)
Memento	객체 정보를 저장 해야 할 때, Undo 기능을 개발 할 때 사용
Chain of responsibility	정적 처리 연결이 하드코딩 되었을 때, 이를 동적으로 연결된 경우에 따라 다르게 처리되도록 연결 (한 요청을 2개 이상의 객체에서 처리)

▼ OSI 7계층

계층	설명	프로토콜	단위
Application	사용자와 네트워크 간 응용서비스 연결, 데이터 생성	HTTP FTP	Data
Presentation	데이터 형식 설정과 부호 교환, 암호/복호화	JPEG MPEG	
Session	연결 접속 및 동기제어	SSH TLS	
Transport	신뢰성 있는 통신 보장. 데이터 분할, 재조립, 흐름 제어, 오류 제어, 혼잡 제어 담당	TCP UDP	Segment
Network	단말기 간 데이터 전송을 위한 최적화된 경로 제공	IP ICMP	Packet
Data Link	인접 시스템 간 데이터 전송, 전송 오류 제어 오류검출/재전송 및 동기화 등 전송 기능 제공	이더넷	Frame
Physical	0과 1의 비트 정보를 회선에 보내기 위한 전기적 신호 변환	RS-232C	Bit

2과목

=====

<<UI 유형>> - CG NO

<<UI 설계 원칙>> - 직유학유

직관성 Intuitiveness	누구나 쉽게 이해하고, 쉽게 사용할 수 있어야 한다.
유효성 Efficiency	정확 하고 완벽하게 사용자의 목표가 달성 될 수 있어야 한다.
학습성 Learnability	누구나 쉽게 배울 수 있어야 한다.
유연성 Flexibility	사용자의 상호작용 을 최대한 포용 하고, 실수를 방지 할 수 있어야 한다.

<<UI 품질 요구 사항>>

▼ ISO/IEC 9126 - 기신사효유이

기능성	적절성, 정확성, 상호운용성, 보안성, 준수성
신뢰성	성숙성, 고장 허용성, 회복성
사용성	이해성, 학습성, 운용성, 친밀성
효율성	시간효율성, 자원효율성
유지보수성	분석성, 변경성, 완전성, 시험성
이식성	적용성, 설치성, 대체성, 공존성

ISO/IEC 25010 : ISO/IEC 9126 에 호환성, 보안성 강화

ISO/IEC 12119 : 패키지 소프트웨어

ISO/IEC 14598 : 품질 평가에 필요 절차를 규정

▼ UI 개발 주요 기법

3C 분석	Customer, Company, Competitor 비교해 경쟁에서 이기도록 분석
SWOT 분석	Strength, Weakness, Opportunity, Threat 요인 규정 및 경영전략 수립
시나리오 플래닝	다양한 시나리오 설계로 불확실성을 제거해 나가는 경영 전략
사용성 테스트	사용자가 시나리오에 따라 직접 제품을 사용 후, 질문에 답함
워크숍	소집단이 새로운 지식, 기술 등을 서로 교환하고 검토하는 세미나

<<UI 화면 설계>> - 와스프

<<UML; Unified Modeling Language>> - 사관다

객체지향 소프트웨어 개발 과정에서 산출물을 명세화, 시각화, 문서화할 때 사용되는 모델링 기술과 방법론을 통합해서 만든 표준화된 범용 모델링 언어

<<UML 특징>> - 가구명문

가시화 언어	개념 모델 작성 시 오류가 적고 의사소통이 용이
구축 언어	다양한 언어로 실행 시스템 예측 가능 UML을 소스코드로 변환하여 구축 가능, 역 변환하면 역공학 가능
명세화 언어	정확한 모델 제시, 완전한 모델 작성 기능
문서화 언어	시스템에 대한 평가 및 의사소통의 문서

<<UML 다이어그램>>

클래스	클래스의 속성, 연산 , 클래스 간 정적 관계 를 표현
객체	인스턴스(객체)를 특정 시점의 객체들 사이 관계로 표현
컴포넌트	시스템을 구성하는 컴포넌트와 그 사이의 의존 관계를 표현
배치	컴포넌트 사이의 종속성 및 프로세스, 컴포넌트 등 물리적 요소 위치 표현
복합체구조	클래스나 컴포넌트가 복합 구조를 가질 때 그 내부 를 표현
패키지	유스케이스, 클래스 등 모델 요소들을 그룹화한 패키지들의 관계 를 표현
유스케이스	시스템 제공 기능 및 관련 외부 요소를 사용자 관점 에서 표현
시퀀스	객체 간 동적 상호 작용을 시간적 개념 중심으로 메시지 흐름 으로 표현
커뮤니케이션	동작에 참여하는 객체들의 메시지 및 연관 을 표현
상태	객체가 속한 클래스의 상태 변화, 상호작용에 따른 상태 변화 를 표현
활동	객체의 처리 로직, 조건에 따른 처리 흐름 을 순서대로 표현
타이밍	객체 상태 변화, 시간 제약 을 명시적으로 표현

▼ UML 스테레오 타입

<<include>>	한 유스케이스가 어떤 시점에 반드시 다른 유스케이스를 실행하는 포함 관계. (주문→사용자확인)
<<extend>>	한 유스케이스가 어떤 시점에 다른 유스케이스를 실행할 수도, 안 할 수도 있는 확장 관계. (고객등록→주문)
<<interface>>	모든 메서드가 추상 메서드이며 바로 인스턴스를 만들 수 없는 클래스. 추상 메서드와 상수만으로 구성된 클래스
<<entity>>	유스케이스 처리 흐름이 수행되는 과정에서 기억 장치에 저장되어야 할 정보 를 표현하는 클래스
<<boundary>>	시스템과 외부 액터와의 상호작용 을 담당하는 클래스
<<control>>	시스템이 제공하는 기능의 로직 및 제어 를 담당하는 클래스

▼ 클래스 다이어그램 - 접근 제어자

+	public	클래스 외부 접근 허용
-	private	클래스 내부 접근만 허용
#	protected	동일 패키지/파생 클래스 접근 가능
~	default	동일 패키지 클래스 접근 가능

<<클래스 다이어그램 관계>> - 연의 일실 포집

Association / Dependency / Generalization / Realization / Composition / Aggregation

3과목

=====

<<데이터 모델 절차>> - 요개논물

<<관계대수>>

관계형 DB에서 원하는 정보와 그 정보를 어떻게 유도하는가를 기술하는 **절차적 정형 언어**

▼ 일반 집합 연산자

합집합 (Union)	\cup
교집합 (Intersection)	\cap
차집합 (Difference)	$-$
카티션 프로덕트	\times

▼ 순수 관계 연산자

선택	σ
프로젝트	π
조인	\bowtie
디비전	\div

<<관계해석>>

프레디캇 해석에 기반한 튜플 / 도메인 관계해석을 하는 **비절차적 언어**

<<논리 데이터 모델링 속성>> - 개속관(ear)

<<정규화 단계>> - 원부이 결다조

<<함수 종속(FD)>>>

Functional Dependency: 릴레이션에서 **속성**의 의미와 상호 관계로부터 발생하는 **제약조건**
 $X \rightarrow Y$ 일 때, X: 결정자 / Y: 종속자 (결정자 \rightarrow 종속자) Y는 X에 종속

부분 함수 종속 (Partial FD)	기본키가 복합키일 경우, 기본키 구성 속성 중의 일부에 종속된 경우
완전 함수 종속 (Full FD)	$X \rightarrow Y$ 일 때, Y는 X의 전체속성에 종속하고, 부분집합 속성에 종속 X
이행 함수 종속 (Transitive FD)	$X \rightarrow Y, Y \rightarrow Z$ 일 때, $X \rightarrow Z$ 가 성립하는 경우

<<반정규화>> - 테병분중 결중 관중

시스템의 성능 향상, 개발 및 운영 편의성 등을 위해 정규화된 데이터 모델을 통합, 중복, 분리하는 과정. 의도적으로 정규화 원칙을 위배하는 행위

<<데이터베이스 무결성>> - 개참속사키

개체무결성 Entity Integrity	한 엔터티에서 같은 PK를 가질 수 없음 PK : NOT NULL	PK Unique Index
---------------------------	---	--------------------

참조무결성 Referential	외래 키가 참조하는 다른 개체의 PK값이 PK이거나 NULL 이어야 함	Foreign Key (Restricted / Cascade)
속성무결성 Attribute	속성 값은 기본값, NULL 여부, 도메인이 지정된 규칙을 준수해야 하는 제약 조건	CHECK, NULL / NOT NULL 기본값(Default)
사용자정의 User-Defined	사용자의 의미적 요구사항을 준수해야 하는 제약조건	Trigger User Defined Data Type
키무결성 Key	한 릴레이션에 같은 키값을 가진 튜플들 을 허용하지 않는 제약 조건	Unique

<<파티셔닝>> - 레해리컴라

<<DBMS 유형>> - 키컬도그

<<Ontology>>

실세계에 존재하는 모든 개념들과 그 속성, 관계 정보를 컴퓨터가 이해하게 한 지식베이스

<<NoSQL; Not Only SQL>>

전통적인 RDBMS와 다른 DBMS를 칭하는 용어. 데이터 저장 시 고정된 **테이블 스키마가 필요하지 않고, 조인 연산을 사용할 수 없으며, 수평적으로 확장이 가능하다.** (BASE)

<<데이터 마이닝>> - 분연 연데

대규모 데이터 안에서 체계적이고 자동적으로 통계적 규칙이나 패턴을 찾아내는 기술

4과목

▼ 직접 연계

DB Link	수신 시스템에서 DB링크를 생성 / 송신 시스템에서 DB링크 직접 참조
DB Connection	수신 시스템의 WAS에서 DB커넥션풀 생성 후 프로그램이 이름 연결
(Open)API	송신 시스템 DB에서 데이터를 읽어 제공. API명, 입출력 파라미터 필
JDBC	수신측 프로그램에서 JDBC 드라이버로 송신측 DB와 연결 (자바 API)
HyperLink	현재 페이지에서 다른 페이지로 이동

▼ 간접 연계

EAI	기업이 운영하는 플랫폼 간 정보 전달, 연계, 통합용 솔루션
Web Service/ESB	WSDL, SOAP를 이용한 시스템 간 연계 (UDDI)
Socket	소켓 생성 후 포트 할당, 클라이언트 요청을 연결해 통신

<<EAI 구축 유형>> - 포터 메타

▼ 웹서비스 관련 용어

HTTP	WWW에서 HTML문서를 송수신하기 위한 표준 프로토콜 인터넷에서 요청과 응답에 의해 처리하는 프로토콜(GET, POST, PUT...)
Hypertext	문장, 단어 등이 링크를 통해 서로 연결된 네트워크처럼 구성된 문서
HTML	웹의 기초적인 구성요소. 웹 콘텐츠의 의미와 구조를 정의할 때 사용 인터넷 웹(WWW)문서를 표현하는 표준화된 마크업 언어

▼ 웹서비스 유형

SOAP	XML 기반 메시지를 네트워크 상에서 교환하는 프로토콜. RPC 패턴 사용 최근에는 더 가벼운 REST 많이 사용
WSDL	웹서비스명, 제공 위치, 메시지 포맷, 프로토콜 정보 등이 기술된 XML 형식언어
UDDI	WSDL을 등록 및 검색하기 위한 저장소로 공개적으로 접근, 검색이 가능한 표준 이자 레지스트리

<<IPC; Inter-Process Communication>>

운영체제에서 프로세스 간 서로 데이터를 주고받기 위한 통신 기술

메시지 큐	메시지 또는 패킷 단위로 동작하여 프로세스 간 통신
공유메모리	한 프로세스의 일부분을 다른 프로세스와 공유
소켓	클라이언트와 서버 프로세스 둘 사이에 통신을 가능하게 함
세마포어	프로세스 사이의 동기를 맞추는 기능을 제공

5과목

<<JSON>>

키-값 쌍으로 이루어진 데이터 객체를 전달하기 위해, 인간이 읽을 수 있는 텍스트를 사용하는 개방형 표준 포맷

<<AJAX>>

JS를 사용해 웹서버와 클라이언트 간 비동기적으로 XML 데이터를 교환하는 웹 기술

<<REST>>

웹과 같은 분산 하이퍼미디어 환경에서 자원의 존재/상태 정보를 표준화된 HTTP 메서드로 주고받는 웹 아키텍처

“(Message)이름이 페코페코인 <사용자(Resource)>를 <생성한다(Method)>” 호출
Resource(URI), Method(GET, POST, PUT, DELETE), Message(XML, JSON) 로 구성됨

<<시큐어 코딩 가이드>> - 입보시 예코캡아

적용대상	보안 약점	대응 방안
입력데이터 검증 및 표현	입력값에 대한 검증 누락, 부적절한 검증, 잘못된 형식 지정	입력 데이터 유효성 검증 체계 수 립 및 실패 시 처리 기능 구현
보안 기능	보안 기능의 부적절한 구현	인증, 접근 통제, 권한 관리, 비밀번호 호 정책이 적절히 반영되도록 설계
시간 및 상태	동시수행 지원 병렬 시스템, 여러 프로세스 동작 환경에서 시간 및 상태의 부적절한 관리	공유 자원 접근 직렬화, 병렬 실행 가능 프레임워크 사용, 블록문 내에 서만 재귀함수 호출
에러 처리	에러 불충분 처리로 에러메시지에 중요 정보가 포함	에러 상황 미처리/불충분 처리로 보 안 약점 발생하지 않도록 구현
코드 오류	개발자가 범할 수 있는 코딩 오류	코딩 규칙 도출 후 검증 가능한 스 크립트 구성, 경고메시지 코드 제거
캡슐화	기능성이 불충분한 캡슐화로 인해 미인가 사용자에게 데이터 유출	디버거 코드 제거, Private 접근자 지정
API 오용	의도와 다른 API 사용, 보안에 취약 한 API 사용	개발 언어별 취약 API 확보 및 검 출 프로그램 사용

<<데이터베이스 암호화 기법 유형>> - 애플티하

<<인터페이스 구현 검증 도구>> - 엑스피 엔셀웨

<<인터페이스 감시 도구>> - 스카우터, 제니퍼

7과목

=====

<<트랜잭션>>

인가받지 않은 사용자로부터 데이터를 보장하기 위해 DBMS가 가져야하는 특성이자, 데이터 베이스 시스템에서 하나의 논리적 기능을 정상적으로 수행하기 위한 작업 기본 단위

▼ 트랜잭션의 특성 - ACID

Atomicity 원자성	트랜잭션이 DB에 모두 반영되거나, 혹은 전혀 반영되지 않아야 한다.	COMMIT/ROLLBACK 회복성 보장
Consistency 일관성	트랜잭션 전후의 DB 상태가 같아야 한다. (DB의 제약, 규칙을 만족해야 한다)	무결성 제약조건 동시성 제어
Isolation 격리성	동시 실행되는 트랜잭션들이 서로 영향을 미치지 않는다.	Read (Un)committed Serializable
Durability 영속성	완료된 트랜잭션 결과는 영구적으로 데이터 베이스에 저장(반영)되어야 한다.	회복기법

▼ 트랜잭션의 상태

활동 상태 Active	초기 상태, 트랜잭션 실행 중 가지는 상태
부분 완료 상태 Partially Committed	마지막 명령문이 실행된 후 가지는 상태
완료 상태 Committed	트랜잭션이 성공적으로 완료된 후 가지는 상태
실패 상태 Failed	정상적인 실행이 더 이상 진행될 수 없을 때 가지는 상태
철회 상태 Aborted	트랜잭션이 취소되고 DB가 트랜잭션 시작 전 상태로 환원된 상태

▼ 병행 제어 기법의 종류 - 로 낙타다

Locking	트랜잭션 실행 중인 데이터에 대해 다른 트랜잭션이 동시 접근하 지 못하도록 상호배제 기능을 제공 (순차진행 보장- 직렬화 기법) - DB, 파일, 레코드 등은 로킹 단위가 될 수 있음 - 로킹단위가 작아지면 DB 공유도, 로킹 오버헤드 증가 - 한꺼번에 로킹할 수 있는 객체의 크기 = 로킹 단위
Optimistic Validation	일단 트랜잭션을 수행하고, 종료 시 검증을 수행하여 DB에 반영
Time Stamp Ordering	트랜잭션 수행 전 타임스탬프를 부여하여 시간에 따라 수행
MVCC; Multi Version Concurrency Control	트랜잭션의 타임스탬프와 접근하려는 데이터의 타임스탬프를 비 교하여 직렬가능성이 보장되는 적절한 버전을 선택하여 접근

▼ 회복 기법 종류 - 회로체그

로그 기반 회복 기법	지연 갱신 회복기법 Deferred Update	트랜잭션 완료 전까지 DB에 기록하지 않음
	즉각 갱신 회복기법 Immediate Update	트랜잭션 수행 중 갱신 결과를 바로 DB에 반영
체크포인트 회복 기법	장애 발생 시 검사점 이후 처리된 트랜잭션에 대해서만 장애 발생 이전 상태로 복원시킴	
그림자 페이지 회복 기법	트랜잭션 수행 시 복제본을 생성하여 DB 장애 시 이를 이용해 복구함	

<<스키마>>

DB의 구조, 제약조건 등의 정보를 담고 있는 기본적인 구조

외부 External	사용자, 개발자 관점에서 필요로 하는 DB의 논리적 구조 (사용자 뷰) = 서브 스키마
개념 Conceptual	DB의 전체적인 논리적 구조 (전체적 뷰) 개체 간 관계, 제약조건, 접근 권한, 무결성, 보안에 대해 정의
내부 Internal	물리적 저장 장치 관점에서 보는 DB 구조 실제 DB에 저장될 레코드 형식 정의, 물리적 순서 등을 표현

▼ 인덱스 종류 - 순해비함 단결클

순서	데이터가 정렬된 순서로 생성되는 인덱스
해시	해시함수에 의해 직접 데이터에 키 값으로 접근하는 인덱스
비트맵	bit 값인 0 또는 1로 변환해 인덱스 키로 사용하는 인덱스
함수기반	수식이나 함수를 적용하여 만든 인덱스
단일	하나의 컬럼으로만 구성된 인덱스
결합	두 개 이상의 컬럼으로 구성된 인덱스
클러스터드	인덱스 키의 순서에 따라 데이터가 정렬되어 저장되는 방식 (검색 빠름)

▼ 다중 행 연산자

IN	리턴 값 중 조건에 해당하는 값이 있으면 참
ANY=SOME	서브쿼리에 의해 리턴되는 각 값과 조건을 비교해 하나 이상 만족할 때 참
ALL	서브쿼리에 의해 리턴되는 모든 값과 조건을 비교해 모든 값 만족 시 참
EXISTS	메인쿼리의 비교 조건이 서브쿼리 결과 중 하나라도 만족하면 참

▼ 절차형 SQL

프로시저	일련의 쿼리들을 하나의 함수처럼 실행하기 위한 쿼리 집합
사용자 정의 함수	일련의 SQL 처리 결과를 단일 값으로 반환할 수 있는 절차형 SQL
트리거	DB 시스템에서 삽삭갱 이벤트 발생 때마다 관련 작업 자동 수행

<<옵티마이저>>

SQL을 가장 빠르고 효율적으로 수행할 최적의 처리경로를 생성해주는 DBMS 핵심 엔진
실행계획 (Execution Plan) : 옵티마이저가 생성한 SQL 처리 경로

8과목

=====

<<개발도구의 분류>> - 빌구데형

웹 서버, WAS, DB 서버, 파일 서버

<<형상관리 절차>> - 식통감기

<<소프트웨어 형상관리 도구 유형>> - 공클분

▼ 결합도 - 내공 외제 스자

내용	직접적으로 다른 모듈 내용 참조
공통	전역변수 를 공유
외부	두 모듈이 외부의 툴, 디바이스 를 공유
제어	다른 모듈의 내부 논리 를 제어하려고 제어 신호 보냄. (권리 전도 현상)
스탬프	배열, 객체, 구조체 를 전달
자료	모듈 간 매개변수(파라미터) 로만 상호작용

▼ 응집도 - 우논시절 통순기

우연적	모듈 내 요소들이 연관성 없이 기능 수행
논리적	논리적으로 유사한 기능이 모임. 밀접하지 않은 관계
시간적	모든 요소들이 같은 시간 에 실행 (초기화 함수 등)
절차적	순서대로 실행. 자료 대신 제어가 전달 됨
통신적	동일한 속성 을 사용하는 기능 묶음. 동일한 입출력 으로 다른 기능을 수행
순차적	한 요소의 출력이 다른 요소의 입력 으로 사용됨 (이어짐)
기능적	모듈의 모든 요소가 단일 기능 만을 수행

<<배치 프로그램 유형>> - 이온정

▼ 배치 스케줄러 종류

Spring Batch	DI, AOP, 서비스 추상화 등 Spring Framework의 3대 요소를 모두 사용할 수 있는 대용량 처리 제공
Quartz Scheduler	Spring Framework에 플러그인되어 수행하는 작업과 실행 스케줄을 정의하는 트리거를 분리해 유연성 제공

<<Cron 표현식>> - 초분시일 월요연

9과목

<<SW 개발 보안의 3대 요소>> - 기무가

기밀성 Confidentiality	인가된 사용자만 자원 접근이 가능함
무결성 Integrity	인가된 사용자만 자원 수정이 가능하며, 전송 중인 정보는 수정 못 함
가용성 Availability	인가된 사용자는 권한 범위 내에서 언제든지 자원 접근이 가능함

<<SW 개발 보안 용어>> - 자위취위

<<DoS; Denial of Service>>

시스템을 악의적으로 공격해 해당 시스템의 자원을 부족하게 해 사용하지 못하게 하는 공격

SYN 플러딩	TCP 프로토콜의 구조적 문제를 이용한 공격 SYN 패킷만 보내 자원이 고갈되도록 함
UDP 플러딩	대량의 UDP 패킷을 임의의 포트번호로 전송해 ICMP를 계속 생성시켜 고갈되도록 함
스머핑	출발지 주소를 공격 대상의 IP로 설정해 네트워크 전체에 ICMP Echo패킷을 직접 브로드캐스팅해 마비시킴. 바운스 사이트를 이용해 공격
죽음의 핑	ICMP 패킷(핑)을 아주 크게 만들어 전송해 단편화를 발생시킴 재조합 버퍼의 오버플로우가 발생해 정상 서비스가 불가능 해짐
랜더택	출발지/목적지의 IP를 같은 패킷주소로 만들어 보내 자기자신에게 응답함
티어 드롭	IP 패킷 재조합 과정에서 Fragment Offset 값을 중첩되도록 조작, 전송해 수신시스템에 문제가 발생하도록 만듦.
붕크	패킷을 보낼 때 순서번호를 모두 1번으로 조작
보잉크	중간에 패킷 시퀀스 번호를 비정상적으로 보내 부하를 일으킴

<<DDoS 공격 구성요소>> - HAMAD

<<DDoS 공격 도구>> Trinoo, Tribe Flood Network, Stacheldraht

▼ 애플리케이션 공격

HTTP GET 플러딩	과도한 Get 메시지를 이용해 웹 서버 과부하를 유발
Slowloris	헤더 끝을 알리는 개행 문자열 대신 \r\n만 전송해 연결자원 소진
RUDY Attack	요청 헤더의 Content-Length를 크게 설정해 바디를 매우 소량 보냄
Slow Read Attack	TCP 윈도우 크기를 낮게 설정해 서버로 전달 후, 연결자원 고갈
Hulk DoS	URL을 지속 변경(우회 목적)하면서 다량의 GET 요청 발생시킴

▼ 네트워크 공격

스니핑	직접 공격 대신 데이터만 몰래 들여다보는 수동 공격
네트워크	네트워크 HW, SW의 취약점 파악을 위해 공격자가 취약점을 탐색하는

스캐너/스니퍼	공격 도구	
패스워드 크래킹	사전	ID, PW 가능성 단어를 파일로 만들어 두고 대입
	무차별	무작위로 PW 대입해 PW 알아냄
	패스워드 하이브리드	사전+무차별 결합
	레인보우 테이블	해시 값을 테이블에서 검색해 역으로 PW 찾음
IP 스누핑	패킷 헤더를 인증된 (척) 호스트의 IP주소로 위조해 타킷에 전송	
ARP 스누핑	MAC주소를 위조해 특정 호스트로 나가는 패킷을 공격자가 스니핑함	
ICMP Redirect	메시지를 위조해 특정 패킷을 공격자가 스니핑함	
트로이목마	겉보기에는 정상이지만 실행하면 악성코드 실행	

▼ 버퍼 오버플로우 공격 대응 방안

스택가드	canary 라고 불리는 무결성 체크용 값을 복귀주소와 변수 사이에 삽입
스택샐드	함수 시작 시 복귀주소를 Global RET라는 특수 스택에 저장 후 비교
ASLR	메모리 공격 방어를 위해 주소 공간 배치를 난수화. 실행 시 마다 메모리 주소를 변경시켜 버퍼 오버플로우를 통한 특정주소 호출 차단
함수	취약: strcpy() strncpy() gets() scanf() sscanf() vscanf() sprintf() vsprintf() 안전: strncat() strncpy() fgets() fscanf() vfscanf() snprintf() vsnprintf()

<<인증 기술의 유형>> - 지소생특

<<서버 접근 통제 유형>>

DAC; 신분 기반 정책 Discretionary Access Control	주체의 신분 에 근거해 객체의 접근을 제한 IBP, GBP (타치고 내맘대로)
MAC; 규칙 기반 정책 Mandatory Access Control	주체가 갖는 권한 에 근거해 객체의 접근을 제한 MLP, CBP (자동으로 등급 따라)
RBAC; 역할기반 정책 Rule Based Access Control	주체가 맡은 역할 에 근거해 객체의 접근을 제한 인사담당자, DBA

3A: Authentication(인증), Authorization(권한부여/인가), Accounting(계정 관리) identification

<<인증 관련 기술>> - SSO, Kerberos

<<접근 통제 보호 모델>> - 벨기비무

<<암호 알고리즘>>

양방향	대칭키 (비밀키)	블록	DES	1975년
			SEED	1999년 한국인터넷진흥원(KISA)
			AES	2001년 미국표준기술연구소(NIST)
			ARIA	2004년 국가정보원+산학연구협회
			IDEA	DES 대체 스위스연방기술기관
	비대칭 (공개키)	스트림	RC4, LFSR, SEAL	
			디피-헬만	
			RSA	
일방향	해시	MAC, MDC	ECC, ElGamal	
			MD5, SHA-1, SHA-256/384/512, HAS-160, HAVAL	

▼ 암호 알고리즘 비교

구분	대칭 키	비대칭 키
키	비밀키	공개키, 개인키
키의 관계	암호화키=복호화키	암호화키 ≠ 복호화키
키 개수	$n(n-1)/2$	$2n$
알고리즘	공개	
장점	계산속도 빠름	암호화키 사전 공유 불필요 관리할 키 개수 적음
단점	키 분배 및 관리 어려움 기밀성만 보장	계산 속도 느림
종류	DES, AES, SEED	디피-헬만, RSA

<<SSL/TLS>>

전송(4)계층과 응용(7)계층 사이 클라이언트와 서버 간의 웹 데이터 암호화(기밀성), 상호 인증 및 전송 시 데이터 무결성을 보장하는 보안 프로토콜

<<자산 목록표>>

자산번호, 자산명, OS, 위치, 담당자(관리자)등의 항목이 존재하며 중요도에 따라 등급 매김

▼ 입력데이터 검증 및 표현 취약점

XSS; Cross Site Scripting	미검증 외부 입력 데이터가 포함된 웹페이지가 전송되는 경우, 사용자가 해당 웹페이지를 열람함으로써 웹페이지에 포함된 부적절한 스크립트가 실행되는 공격	- 특수문자 필터링 - HTML 태그 사용 금지 - JS로 시작하는 문자열은 문자열 변환처리
사이트 간 요청 위조 CSRF; Cross Site Request Forgery	사용자가 의지와 무관하게 공격자가 의도한 행위를 특정 웹사이트에 요청하게 하는 공격	- POST 방식 사용 - 세션별 CSRF 토큰 사용 - 재인증으로 요청 확인
SQL 삽입	응용 프로그램의 취약점을 이용해 악의적인 SQL 구문 삽입, 실행시켜 DB에 접근해 정보 탈취 및 조작	바인딩 매개변수 방식을 적용해 변수 타입 명시 지정 Servlet Filter 기능 적용

<<네트워크 보안 솔루션>>

방화벽 Firewall	기업 내/외부 간 트래픽을 모니터링하여 시스템의 접근을 허용하거나 차단
웹 방화벽 WAF	웹 애플리케이션 보안에 특화 SQL 삽입, XSS 같은 웹 공격을 탐지하고 차단
네트워크 접근 제어 NAC	단말기가 내부 네트워크에 접속을 시도할 때 이를 제어하고 통제하는 기능을 제공하는 솔루션
침입 탐지 시스템 IDS	네트워크에 발생하는 이벤트를 모니터링하고, 침입을 실시간으로 탐지하는 시스템
침입 방지 시스템 IPS	네트워크에 대한 공격이나 침입을 실시간으로 “자동” 차단하는 시스템

무선침입방지 시스템 WIPS	무선 단말기의 접속을 자동 탐지/차단
통합 보안 시스템 UTM	다양한 보안 장비의 기능을 하나로 통합한 장비 (보통 IPS+IDS+방화벽)
가상사설망 VPN	인터넷 같은 공중망에 인증, 암호화, 터널링 기술을 활용해 마치 전용망을 사용하는 효과를 가지는 보안 솔루션
SIEM	기업에서 생성, 수집되는 다양한 데이터 분석을 통해 보안 위험 징후를 빠르게 판단하고 대응하는 보안 관제 솔루션
ESM	보안 장비들을 통합 관리하는 기능 및 네트워크 보안 모니터링, 이벤트 위주 단시간 위협 분석 및 DBMS 기반 보안 관리 솔루션

<<취약점 분석 절차>> - 자진 제진결보

<<비즈니스 연속성 계획; BCP>>

각종 재해, 장애, 재난으로부터 위기관리를 기반으로 재해 복구, 업무복구 및 재개, 비상 계획 등을 통해 비즈니스 연속성을 보장하는 체계

BCP를 위한 비즈니스 영향 평가(BIA)가 선행되어야 한다.

▼ BCP 주요 용어

BIA	장애, 재해로 인한 주요 손실을 가정하여 시간 흐름에 따른 영향도, 손실 평가를 조사하는 BCP를 구축하기 위한 비즈니스 영향 분석
RTO	업무중단 시점부터 복구되어 다시 가동될 때까지의 시간 재해 시 복구 목표 시간
RPO	업무중단 시점부터 복구되어 다시 가동될 때 데이터의 손실 허용 시점 재해 시 복구 목표 지점
DRP	재난으로 장기간 운영이 불가능한 경우를 대비한 재난 복구 계획
DRS	DRP의 수행 지원을 위해 평상 시 확보해두는 인적, 물적 자원 및 지속적인 체계 관리가 통합된 재해복구센터

▼ DRS 유형

Mirror Site	주센터, 데이터복구센터 모두 운영 상태로 실시간 동시 서비스가 가능 RTO : 즉시(0)
Hot Site	주센터와 동일 수준의 자원을 대기 상태로 원격지에 보유해 미러링을 통해 최신 상태를 유지. RTO : 4시간 이내
Warm Site	핫사이트와 유사하지만 중요성 높은 자원만 부분적으로 보유 RTO : 수일 ~ 수주
Cold Site	데이터만 보관하다 재해 시 필요 자원을 조달하여 복구 RTO : 수주 ~ 수개월 (구축비용은 저렴하나 RTO가 길고 신뢰성 낮음)

10과목

=====

<<소프트웨어 테스트의 원리>> - 결완초집 살정오

결함 존재의 증명	결함이 존재함을 밝힘 결함을 줄임
완벽 테스트 불가능	완벽한 테스트 시도는 불필요한 시간, 자원 낭비
초기 집중	초기 분석 및 설계가 수행되지 않으면 프로젝트 후반에 영향을 미침 요르돈의 법칙(눈덩이 법칙)
결함 집중	소프트웨어 테스트에서 오류의 80%는 전체 모듈의 20% 내에서 발견 파레토 법칙
살충제 패러독스	동일한 내용의 반복적 테스트로는 새로운 오류를 찾을 수 없음
정황 의존성	소프트웨어 성격에 맞게 테스트 실시 정황과 비즈니스 도메인에 따라 테스트 다르게 수행
오류 부재의 퀘변	요구사항을 충족하지 못하면, 결함이 없어도 품질이 높다고 할 수 없음

<<화이트박스 테스트 기법>> - 구결조 조변다 기제테루

구조 기반 테스트, 모듈 내부 소스 코드를 보면서 수행하는 테스트

구문 커버리지 Statement Coverage	프로그램 내 모든 명령문 을 적어도 한 번 수행 (조건문 결과와 관계 없이 구문 실행 개수로 계산)
결정 커버리지 Decision Coverage	결정 포인트 내의 전체 조건식 이 적어도 한 번은 참과 거짓 결과를 수행
조건 커버리지 Condition Coverage	결정 포인트 내의 각 개별 조건식 이 적어도 한 번은 참과 거짓 결과를 수행
조건/결정 커버리지 C/D Coverage	전체 조건식 뿐만 아니라 개별 조건식도 참 한 번, 거짓 한 번 결과가 되도록 수행
변경 조건/결정 커버리지 Modified C/D Coverage	개별 조건식이 다른 조건식에 영향을 받지 않고 전체 조건식에 독립적으로 영향 을 주도록 함으로써 위를 향상시킴
다중 조건 커버리지 Multiple C Coverage	결정 조건 내 모든 개별 조건식의 모든 가능한 조합 을 100% 보장
기본 경로 커버리지 Base Path Coverage	수행 가능한 모든 경로 테스트
제어 흐름 테스트 Control Flow Testing	프로그램 제어구조를 그래프 형태 로 나타낸 내부 로직테스트
데이터 흐름 테스트 Data Flow Testing	제어흐름 그래프에 데이터 사용 현황 을 추가
루프 테스트 Loop Testing	프로그램 루프 구조 에 초점을 맞춰 실시

<<블랙박스 테스트 기법>> - 동경결상 유분페원비오

명세 기반 테스트, 외부 사용자의 요구사항 명세를 보면서 수행하는 기능 테스트

Equivalence Partitioning Testing	입력 데이터 영역을 유사한 도메인별로 유/무효값을 그룹핑하여 대푯값 테스트 케이스를 도출
Boundary Value Analysis Testing	입력값의 경계를 테스트하는 기법
Decision Table Testing	요구사항의 논리, 발생조건을 테이블 형태로 나열해 조건과 행위를 모두 조합해서 테스트
State Transition Testing	테스트 대상의 상태를 구분하고 이벤트에 의해 다른 상태로 전이되는 경우의 수를 수행
Use Case Testing	시스템이 유스케이스로 모델링 됐을 때 프로세스 흐름을 기반으로 테스트 케이스를 명세화하여 수행
Classification Tree Method Testing	SW의 일부/전체를 트리 구조로 분석 및 표현하여 테스트케이스를 설계
Pairwise Testing	테스트 데이터값들 간 최소 한 번씩을 조합함. 커버할 범위 조합을 상대적으로 적게 테스트 세트 구성
Cause-Effect Graph Testing	그래프를 활용해 입력 데이터 간의 관계 및 출력에 미치는 영향을 분석해 효용성 높은 케이스 선정
Comparison Testing	여러 버전의 프로그램에 같은 입력값을 넣어 동일한 결과가 나오는지 비교
Error Guessing Testing	개발자가 범할 수 있는 실수를 추정해 이에 따른 결함이 검출되도록 케이스를 설계

▼ 테스트 시각에 따른 분류

검증	SW 개발 과정 테스트, 올바르게 생산하는지 검증
Verification	개발자/시험자 시각으로 SW가 명세화된 기능을 올바르게 수행하는지 알아봄
확인	SW 결과 테스트, 완성품이 제대로 동작하는지 확인
Validation	사용자 시각으로 올바른 SW가 개발되었는지 입증하는 과정

<<테스트 목적에 따른 분류>> - 회안성 구회병

Recovery	고의로 실패 유도 후, 시스템의 정상적 복귀 여부를 테스트
Security	소스 코드 내의 보안 결함을 미리 점검하는 테스트
Performance	시스템 응답 시간, 처리량, 반응 속도 등을 측정하는 테스트
Structure	시스템의 내부 논리 경로, 소스 코드의 복잡도를 평가하는 테스트
Regression	오류 제거, 수정에 의해 새로 유입된 오류 여부를 확인하는 반복 테스트
Parallel	변경된 시스템과 기존 시스템에 동일한 데이터 입력 후 결과 비교

<<성능 테스트 유형>> - 부장스내

▼ 테스트 오라클 - 참샘휴일

참 오라클	모든 입력값에 대해 기대하는 결과를 생성해, 오류를 모두 검출
샘플링	특정 몇 개 입력값에 대해서만 기대한 결과를 제공
휴리스틱	샘플링 개선 방식. 나머지값들에 대해 휴리스틱(추정) 으로 처리
일관성검사	애플리케이션 변경 시, 수행 전 후의 결과값이 동일한지 확인

▼ 테스트 자동화 도구

정적 분석 도구	애플리케이션을 실행하지 않고 분석하는 도구 소스 코드에 대한 이해를 바탕으로 분석 / 소스 코드 결함 발견
테스트 실행 도구	테스트용 스크립트를 실행 데이터 주도 접근 방식, 키워드 주도 접근 방식 존재
성능 테스트 도구	가상의 사용자를 생성해 성능 목표를 달성했는지 확인
테스트 통제 도구	테스트 계획 및 관리를 위한 도구. 다른 도구와 연계해서 사용

<<테스트 하네스>> - 드 스슈 케이스목
테스트를 지원하기 위한 코드 및 데이터

<<테스트 커버리지>> - 기라코
주어진 테스트케이스에 의해 수행되는 SW의 테스트 범위를 측정하는 테스트 품질 측정기준
테스트의 정확성, 신뢰성을 향상시킴

<<애플리케이션 성능 측정 지표>> - 처응경자

11과목

<<메모리 관리 기법>> - 반배할교

▼ 주기억장치 할당 기법 - 연단다 분배세

연속할당기법	프로세스를 주기억장치에 연속으로 할당	단일분할 할당기법: 오버레이, 스와핑 다중분할 할당기법: 고정분할, 동적분할
분산할당기법	프로세스를 여러 개로 나눠 분산하여 배치(가상기억장치)	페이징, 세그먼테이션 페이징/세그먼테이션

▼ 교체 기법 유형

FIFO First In First Out	가장 오래된 것부터 교체
LRU Least Recently Used	가장 오래 사용되지 않은 것부터 교체
LFU Least Frequently Used	사용 횟수가 가장 적은 것부터 교체
OPT Optimal Replacement	앞으로 오래 사용하지 않을 것부터 교체 페이지 부재 횟수가 가장 적게 발생(효율적)
NUR Not Used Recently	최근에 사용하지 않은 것부터 교체 LRU의 시간적 오버헤드 감소
SCR Second Chance Replacement	가장 오래 있으면서 자주 사용되는 페이지 교체 방지 FIFO 단점 보완

<<메모리 단편화>>

분할된 주기억장치에 프로세스를 할당, 반납과정에서 사용되지 못하고 낭비되는 기억장치가 발생하는 현상

내부 단편화	분할된 공간에 프로세스 적재 후 남은 공간 (고정 분할 할당 방식 / 페이징 기법 사용 시 발생)
외부 단편화	할당된 크기가 프로세스 크기보다 작아서 사용하지 못하는 공간 (가변 분할 할당 방식 / 세그먼테이션 기법 사용 시 발생)

<<프로세스 상태 전이>> - 생존 실패완, 디타 블웨, Context Switching

▼ 비선점 스케줄링 - 우기HFS

Priority; 우선순위	우선순위에 따라 CPU 할당 (동일 순위는 FCFS)
Deadline; 기한부	작업들이 명시된 기한 내에 완료되도록 계획
FCFS; First Come First Serve	먼저 온 순서대로 CPU 할당

SJF; Shortest Job First	실행시간이 가장 짧은 것부터 CPU 할당
HRN; Highest Response-ratio Next	우선순위 결과 값이 높은 것부터 CPU 할당 대기시간이 길 경우 결과 값이 높아짐. (SJF 보완 기법) $\text{결과값} = (\text{대기시간} + \text{서비스시간}) / \text{서비스시간}$

▼ 선점 스케줄링 - SMMR

RR; Round Robin	각 프로세스를 할당된 시간 동안 실행 후, 실행이 완료되지 않으면 다음 프로세스에 CPU를 넘기고 큐의 맨 뒤에 배치
SRT; Shortest Remaining Time	SJF를 선점형태로 변형. 현재 실행 중인 프로세스의 남은 시간과 새로 도착한 프로세스의 실행시간을 비교해 더 짧은 실행 시간을 요구하는 프로세스에 CPU 할당
MLQ; Multi Level Queue	프로세스를 특정 그룹으로 분류할 수 있을 경우, 그룹에 따라 각기 다른 준비상태 큐를 사용
MLFQ; Multi Level Feedback Queue	MLQ에서 다른 큐로 이동할 수 있도록 개선

<<교착상태 발생 조건>> - 상점비환

<<교착상태 해결 방법>> - 예회발복, 은행가 알고리즘

<<원거리 통신망(WAN) 연결 기술>> - 전화패

<<디스크 스케줄링>>

사용할 데이터가 디스크 상의 여러 곳에 저장된 경우, 데이터를 액세스하기 위해 디스크 헤드를 움직이는 경로를 결정하는 기법. OS가 담당. 처리량 최대화, 응답시간 최소화 목적.

FCFS	디스크 대기 큐에 먼저 들어온 요청부터 서비스
SSTF	현재 위치에서 탐색거리가 가장 짧은 트랙 요청부터 서비스 일괄처리 시스템에 유용 (가장 가까운 트랙부터)
SCAN	현재 헤드에서 진행 방향이 결정되면 탐색거리가 짧은 순서에 따라 그 방향의 모든 요청 처리, 끝까지 이동 후 역방향 요청 처리
C-SCAN	항상 바깥→안쪽으로 움직이며 가장 짧은 탐색거리 요청부터 서비스 안쪽 끝까지 가면 다시 바깥쪽부터 탐색함: 비교적 공평
LOOK	SCAN은 이동방향 끝까지 가지만 LOOK은 요청까지만 갔다 방향 전환 =엘리베이터 알고리즘
N-STEP SCAN	방향 진행이 시작될 때 대기 중이던 요청들만 처리, 진행 중 도착한 요청들은 썩 모아서 역방향으로 진행할 때 처리
SLTF	=Sector Queuing 회전지연시간 최적화 실린더 내 여러 요청들을 검사 후, 회전지연시간이 가장 짧은 것부터

<<클라우드 컴퓨팅 분류>> - 사공하

<<클라우드 서비스 유형>> - 인플소

<<네트워크 장비>>

▼ 1계층 장비

허브	여러 대의 컴퓨터를 연결하여 네트워크로 보내거나 하나의 네트워크로 수신된 정보를 여러 대의 컴퓨터로 송신하기 위한 장비
리피터	디지털 신호를 증폭시켜 주는 역할을 해 신호가 약해지지 않고 컴퓨터로 수신 되도록 하는 장비

▼ 2계층 장비

브리지	두 근거리 통신망(LAN)을 연결해주는 통신망 연결 장치
L2 스위치	출발지에서 들어온 프레임의 목적지 MAC 주소 기반으로 빠르게 전송하는 데이터 링크 계층 통신장치 Store and Forwarding / Cut Through / Fragment Free VLAN / STP
NIC	외부 네트워크와 접속해 가장 빠른 속도로 데이터를 주고받도록 컴퓨터 내부에 설치
스위칭 허브	스위치 기능을 가진 허브(대부분 허브는 스위칭 허브)

▼ 네트워크 계층(3계층) 프로토콜

IP; Internet	송수신 간 패킷 단위로 데이터를 교환하는 네트워크에서 사용하는 통신 규약
ARP; Address Resolution	IP 네트워크 상에서 IP주소를 MAC주소로 변환하는 프로토콜
RARP; Reverse ARP	IP 호스트가 자신의 MAC주소(물리주소)만 알 때, 서버로부터 IP주소를 요청하기 위해 사용하는 프로토콜
ICMP Internet Control Message	IP 패킷을 처리할 때 발생하는 에러를 알려주는 프로토콜 (Ping 이 ICMP 패킷을 통해 테스트 함)
IGMP Internet Group Management	호스트 컴퓨터와 인접 라우터가 멀티캐스트 그룹 멤버쉽을 구성하는 데 사용하는 프로토콜
Routing Protocol	데이터 전송을 위해 목적지까지 갈 수 있는 여러 경로 중에 최적 경로를 설정해주는 라우터 간의 상호 통신 프로토콜

▼ IPv4 / IPv6

구분	IPv4	IPv6
주소 길이	32Bit	128Bit
표시 방법	8비트씩 4부분으로 나뉜 10진수	16비트씩 8부분으로 나뉜 16진수
주소 개수	약 43억 개	3.4×10^{38}
주소 할당	클래스 단위 비순차적 할당 (비효율)	네트워크 규모 및 단말기 수에 따른
헤더 크기	20byte 기본 헤더+가변 길이 옵션	40byte 고정 길이
QoS	Best Effort 방식 (보장 곤란)	등급별, 서비스별 패킷 구분 보장
보안 기능	IPSec 프로토콜 별도 설치	보안, 인증확장 헤더 사용
Plug&Play	지원 안함	지원
모바일 IP	곤란	용이
웹 캐스팅	곤란	용이
전송 방식	유니캐스트, 멀티캐스트, 브로드캐스트	유니캐스트, 멀티캐스트, 애니캐스트

<<내부 라우팅 프로토콜(IGP; Interior Gateway Protocol)>>

* RIP(Routing Information Protocol): AS(자율시스템)내에서 사용하는 거리 벡터 알고리즘에 기초하여 개발된 내부 라우팅 프로토콜

* OSPF(Open Shortest Path First): 자신을 기준으로 링크 상태 알고리즘을 적용하여 최단 경로를 찾는 라우팅 프로토콜

<<외부 라우팅 프로토콜(EGP; Exterior Gateway Protocol)>>

* BGP(Border Gateway Protocol): AS상호 간에 경로 정보를 교환하기 위한 라우팅 프로토콜

▼ 응용 계층(7계층) 프로토콜

HTTP	80	텍스트 기반, 인터넷에서 데이터(hypertext) 교환
FTP	21	TCP/IP 프로토콜로 서버와 클라이언트 사이 파일을 전송
SMTP	25	TCP 포트 번호 25번을 사용해 이메일을 보내기 위해 사용
POP3	110	원격 서버에서 TCP/IP 연결을 통해 이메일을 가져오는 데 사용
IMAP	143	원격 서버에서 TCP/IP 연결을 통해 이메일을 가져오는 데 사용
Telnet	23	인터넷/로컬 영역에서 네트워크 연결에 사용
SSH	22	텔넷보다 강한 보안성. 인증,암호화,압축,무결성 제공
SNMP	161	라우터, 허브 등 네트워크 장치로부터 정보 수집 및 관리 시스템 전송

<<스토리지 시스템 유형>>

DAS Direct Attached Storage	저장 장치와 호스트 기기 사이 직접 연결
NAS Network Attached Storage	서버와 저장 장치 사이 네트워크 연결 파일 단위로 관리
SAN Storage Area Network	블록 단위로 관리, 스토리지 공유 가능 DAS의 빠른 처리 + NAS의 스토리지 공유 방식 합침

<<RAID; Resundant Array of Inexpensive/Independent Disk>>

2개 이상의 하드디스크로 디스크 배열 구성. 데이터 블록을 서로 다른 디스크에 분산 저장하거나 다중화

RAID 0	패리티(오류 검출 기능)가 없는 스트라이핑된 세트로 구성 최소 2개 필요 / 하나만 손상되어도 전체 디스크 파손
RAID 1	다른 디스크에 동일하게 복사하는 미러링 구성 최소 2개 필요 / 비용 측면에서 단점
RAID 2	하나의 디스크에 오류정정부호(ECC)를 비트 단위로 저장 최소 3개 필요 / 입출력 속도 매우 느림 / 디스크 하나 손상되어도 정상 가동
RAID 3	하나의 디스크에 오류정정부호(ECC)를 바이트 단위로 저장 최소 3개 필요 / 읽기 성능은 좋지만 문제 해결이 어려움
RAID 4	하나의 디스크에 오류정정부호(ECC)를 워드 단위로 저장 최소 3개 필요 / 읽기 성능은 좋지만 쓰기 성능은 나쁨
RAID 5	여러 개의 디스크에 ECC를 분산 저장 / 최소 3개 필요
RAID 6	여러 개의 디스크에 ECC 2개를 분산 저장 / 최소 4개 필요 디스크 2개 손상되어도 정상 가동

12과목

=====

<<릴리즈 노트 작성 항목>> - 해개목이 재수사소 노면언

▼ 패키징 도구 구성 세부 기술

PKI	인증기관에서 공개키 암호 방식의 전자서명된 인증서를 발급받아 네트워크 상에서 안전하게 비밀통신을 가능하도록 하는 기술
대칭/비대칭 암호화	대칭 암호화: 암호화키 = 복호화키 비대칭 암호화: 암호화키 ≠ 복호화키
전자서명	서명자를 확인하고 서명자가 해당 전자문서에 서명했다는 사실을 나타내기 위해 특정 전자문서에 첨부되거나 논리적으로 결합된 정보
DOI Digital Object Identifier	디지털 저작물의 저작권 보호 및 정확한 위치 추적을 위해 특정한 번호를 부여하는 바코드 시스템
URI	인터넷에 있는 자원을 고유하게 식별할 수 있도록 나타낸 주소 URL+URN
XrML	디지털 콘텐츠/웹 서비스 권리 조건을 표현한 XML 기반 마크업 언어
MPEG-21	멀티미디어 관련 요소 기술들이 통일된 형태로 상호 운용성을 보장하는 멀티미디어 표준 규격
Pre-packaging	콘텐츠를 등록하자마자 암호화
On-the-fly Packaging	사용자가 콘텐츠를 요청한 시점에 콘텐츠를 암호화
XML	다른 특수 목적을 갖는 마크업 언어를 만드는 데 사용하는 다목적 언어
CMS	다양한 미디어 포맷에 따라 콘텐츠 생성부터 활용, 폐기까지 전 공급 과정을 관리
코드 난독화	역공학 공격을 막기 위해 프로그램 소스 코드를 알아보기 힘들게 변경
Secure DB	커널 암호화 방식으로 DB 파일을 직접 암호화. 접근 제어와 감사 기록 기능이 추가된 DB 보안 강화 기술
SSO	한 번의 시스템 인증을 통해 여러 정보시스템에 재인증 절차 없이 접근할 수 있는 통합 로그인 기술

<<백업 유형>> 전차증