

<c언어>

1. `Printf("%s", &b[6]);`

1. 0~5인덱스 무시. 6~끝 까지의 문자열 출력

2. 랜덤값

1. `rand()%6 + 1`

2. 1~7사이 난수 발생. 즉, 0~6 사이 인덱스 맞출려면 `h[n-1]` 해야 맞음.

3. 아스키코드

1. `65 = A`

2. `97 = a`

3. `char c = 65 -> printf("%c", ++c) = B`

4. 포인터

1. 주소연산자 = `&`

1. 해당 변수의 주소값 반환

2. 참조 연산자 = `*`

1. 포인터가 가르키는 주소에 저장된 값

3. `*&`가 동시에 있으면 생략이 가능

4. 배열

1. `&array[0]` => array배열 0번째의 주소 => 0번째가 `int` 라면 요소의 크기는 4 바이트 출력

2. `p`가 포인터일때, `int *p = &x` 이면

3. `*(p+2) = x[2]`

5. 2차원 배열

1. 2차원 배열은 `*` 이 두번 붙어야 값이다. 즉, `*` 이 하나면 주소를 뜻한다.

2. `*(p[0] + 1) == (*(p + 0) + 1)` => `p[0]` 주소 +1 한 값 -> `p[0][1]` 의 값

5. C++

1. `count << "b" -> b`출력

2. `delete a -> ~a()` 실행

<자바>

1. 상속관계 호출

1. 부모에서 `print()`호출해도 자식의`print()` 가 호출됨

2. 16진수 계산

1. `%x` = 16진수 표기법 (10진수를 16진수로 변환)

2. $52 \% 16 = 3 \dots 4$ (나머지)
3. $3(\text{몫}) \% 16 = 0 \dots 3$ (나머지)
4. 따라서 51 의 16진수 -> 34
3. 랜덤값
 1. `Math.random()*10`; //0 ~ 10 사이
 2. `Math.random()`은 0.0~ 1.0 사이난수 발생
4. 배열
 1. 행이0~2, 열이 0~4 가지는 구조라면, `a[3][5]` 가 필요

<파이썬>

1. 리스트 연산
 1. `append(10)` -> 리스트 맨 끝에 10 추가
 2. `remove(2)` -> 리스트 값중 숫자2 삭제
 3. `a[2]` -> 인덱스 2번의 값 선택 (리스트는 0인덱스부터 시작)
2. `count(값)` -> 값의 요소 수를 반환
3. `index(10)` -> 10의 위치를 반환.
4. `copy()` -> 깊은복사해 각각의 메모리 주소를 가지게 한다.
5. `pop(위치)` -> 위치에 있는 값 출력 후 요소 삭제
6. `set의 pop()` -> 세트의 값 출력인데, 순서 모름
7. 반복문
 1. `range(시작 숫자, 끝 숫자)` -> 끝 숫자 포함 안됨
 2. 즉, `range(1,11)` 이면 1~10까지 숫자 반복임.
 3. `c = [a[i] + b[i] for i in range(4)]` -> 0~3 까지 숫자 반복해 c의 배열 [0,1,2] 번째에 들어감
8. 합계
 1. `sum(a)` : 배열 또는 리스트 전체의 합 구하는 함수
9. 리스트 더하기
 1. `append()` : 맨뒤에 더하기
 2. `insert(index,값)` : 원하는 위치에 값넣기
10. `+`: 리스트에 값 추가

<sql>

릴레이션

행, 튜플, 레코드 행의 수 = 카디널리티, 튜플 수

열, 속성, 필드 열의 수 = 차수, 디그리

1. 참조 데이터 제거
 1. `drop view A RESTRICT` : A개체 참조하는 모든 개체들 삭제
 2. `drop view A CASCADE` : A개체 참조하고 있는 개체 있다면 명령 취소
2. 연산함수
 1. 집합함수 `count()`, `sum()`,`avg()` 등은 null 갯수 포함 안함
3. 레코드 수

1. select count(*) 한 레코드의 수는 1이다.
2. count(*) 과 4 값만 나오니까 레코드는 1이다.

4. 필드명

1. 무조건 as 쓰기.
2. a.자격증명 -> a.자격증명 as 자격증명

5. IN() 사용

1. select * from A where 학번 IN(3,4)
2. 3,4 학년만 출력

6. 중복제거

1. select distinct 학년 from 학생

7. 인덱스 스키마 생성

1. create index 인덱스명 on 테이블명(속성명)

8. 순위

1. select rank(점수) : 2위,2위,2위,5위 -> 중복이면 그만큼 순위 밀림
2. select row_number(점수) : 1위,2위,3위 -> 중복관계없이
3. select DENSE_RANK() OVER (ORDER BY 점수 DESC) as 점수 : 2위,2위, 3위 -> 중복인 경우 같은 순위

<DDL>

1. 데이터 구조를 정의하는데 사용
2. Create
 1. create index 인덱스명 ON dbo.emp (hiredate)
 2. create view 뷰이름 as (SELECT FROM 구문)
 3. create table A(emp_id NUMBER NOT NULL)
3. Drop
 1. drop [스키마, table] 테이블명
4. Alter table
 1. Alter table 테이블명 ADD 컬럼명
 2. alter table 테이블명 modify 컬럼명 NUMBER(6)
 3. alter table 테이블명 modify A INTEGER PRIMARY KEY

<DCL> - Data Control Language

1. 데이터의 보안, 무결성, 회복, 병행 수행 제어 등을 정하는데 사용하는 언어
2. Commit
 1. commit
3. Rollback
 1. rollback to 세이브포인트
4. Revoke
 1. revoke 등급 on 테이블명 from 사용자

5. Grant

1. Grant 사용자등급 on 테이블명 to 사용자
2. Grant ALL ON A TO 사용자 WITH GRANT OPTION -> 다른 사람에게 권한 줄수있도록 권한부여

<DML> - Data Manipulation Language

1. 사용자가 실질적으로 저장된 데이터를 처리 할때 사용
2. select * from A
3. Insert into A values (a,b,...)
4. Delete from A where..
5. Update A set 주소 = a

<TCL> 트랜잭션 컨트롤 언어

1. Commit : 트랜잭션을 메모리에 영구적 저장
2. Rollback : 트랜잭션 내역을 무효화
3. Checkpoint : 롤백위한 시점


<합집합>

1. Union : 합집합. 두 select문 조회결과 통합해 모두 출력 (중복행 하나만 출력)
2. Union all : 두 select문 조회결과 통합해 모두 출력 (중복행 그대로 출력)
3. Insect : 두 select문 조회결과 중 공통행만 출력
4. Except : 첫select문 조회결과에서 두번째select문 결과 제외한것만 출력

<Join>

1. Left Outer : 왼쪽 모든데이터, 오른쪽 일치 데이터만 출력(왼쪽 출력시 오른쪽 필드 값 없는 경우 Null처리)
2. Right Outer : 오른쪽 모든 데이터, 왼쪽 일치 데이터만 출력
3. Full Outer : 양쪽 모든 데이터 출력

<문자열 찾기 like >

1. % : 0개 이상 문자열과 일치
2.  : 1개의 문자와 일치
3. [^] : 1개의 문자와 불일치
4. _ : 특정위치 1개 문자와 일치

<관계 대수>

- 순수 관계 연산자
- 셀렉트, 파이, 조인, 디비전
 1. 셀렉트(시그마) : 선택 조건 만족하는 튜플을 새 릴레이션으로 만드는 연산 .가로행 가져오기.
 2. 프로젝트(파이) : 제시된 속성값만 추출하여 새 릴레이션으로 만드는 연산. 중복 발생시 제거. 세로 행 가져오기
 3. 조인() : 두 릴레이션 합쳐 새 릴레이션 생성. 중복 허용
 4. 디비전(÷) : R과 S 릴레이션이 있을때 R의 속성이 S의 속성값을 가진 튜플에서 S가 가진 속성 제외한 속성을 구하는 연산.

- 일반 관계 연산자
 - 카디션곱, insertion, 유니온, difference
 - 카디션곱 : 교차곱으로 차수(열)는 더하고 카디널리티(행)은 곱한다. 문자 X사용.

<연산자 순위>

- 1. 증 산 시 관 비 논 조 대 순
- 2. 감 술 프 트 게 트 리 건 입 서

<자료사전 기호>

- 1. = : 정의
- 2. ◦ : 연결
- 3. () : 생략
- 4. { } : 반복
- 5. □ : 선택
- 6. ** : 주석

<UML>

- 객체지향 모델링 언어로 개발시 사용자와 원활한 의사소통 하기 위해 사용

정적 다이어그램

객체지향 모델링시 정적인 관계 표현	클래스
인스턴스를 특정 시점의 객체와 객체 사이의 관계 표현	객체
클래스가 복합구조 가질때 내부 구조 표현	복합
컴포넌트 사이 종속성 표현과 물리적 요소 위치 표현	배치
의존관계 나타냄	컴포넌트
클래스 모델 요소들 그룹화	패키지

동적 다이어그램

사용자 측면에서 요구분석해 기능 모델링 작업에 사용	유스케이스
상호작용하는 시스템이나 객체들이 주고받는 메시지 표현	시퀀스
객체들간 주고받는 메시지와 객체간의 관계까지 표현	커뮤니케이션
객체가 자신이 속한 클래스의 상태변화에 따라 어떻게 변하는지 표현	상태
객체 처리 로직이나 조건에 따른 처리 흐름 순서에 따라 표현	활동
객체 상태 변화와 시간 제약 명시적으로 표현	타이밍
상호작용 다이어그램 간의 제어 표현	상호작용

uml 관계 종류

2개 이상의 사물이 서로 관련되어 있음을 표현	연관관계
---------------------------	------

uml 관계 종류	
하나의 사물이 다른 사물에 포함	집합관계
집합관계의 특수한 형태로, 포함하는 사물의 변화가 포함되는 사물에 영향	포함관계
하나의 사물이 다른 사물에 비해 일반적인지 구체적인지 표현	일반화 관계
사물 사이에 연관은 있으나 필요에 의해서 서로에게 영향주는 짧은 시점에만 관계성립	의존관계
사물이 할 수 있거나 해야하는 기능으로 서로를 그룹화	실체화 관계

<GOF 디자인 패턴 >

- 어댑터 vs 프록시
 - 어댑터는 인터페이스 이용
 - 프록시는 접근 제어자 이용
- 파싸드 vs 템플릿 메소드
 - 파싸드는 상위 구현, 하위 사용
 - 템플릿 메소드는 상위 골격, 하위 구현

생성	
구체적 클래스에 의존하지 않고 인터페이스 통해 연관있는 객체들의 그룹으로 생서해 추상적으로 표현	추상
인스턴스를 조합하여 객체를 생성	빌더
객체 생성을 서브 클래스에서 구현하도록 분리하여 캡슐화	팩토리
비용이 큰 경우 사용하는 패턴으로, 원본 객체를 복사하는 방법으로 객체 생성	프로토 타입
클래스 내에서 인스턴스가 하나뿐임을 보장하는 패턴, 하나 객체 생성하면 어디서든 해당 객체 참조가능	싱글톤

구조	
다른 클래스 연결되는 패턴. 인터페이스 이용 해 클래스 변환.	어댑터
기능과 구현을 두개의 별도 클래스로 구현해 서로 독립적으로 확장할 수 있도록 함	브릿지
복합객체와 단일객체를 구분없이 다루고자 할 때 사용	컴포지트
객체간의 결합을 통해 기존 클래스에 필요한 기능 추가	데코레이터
상위 인터페이스에 기능 구현하고 하위에서 쉽게 사용	파싸드
메모리 절약을 위해 인스턴스 가능한 공유해서 사용	플라이위이드
다른 클래스 연결 돕는 패턴. 특정 객체로 접근 제어하는 대리자 제공	프록시

행위	
요청처리 못하면 다음 객체가 처리	책임연쇄

행위	
각종 명령들을 분리해 단순화	커맨드
언어문법 표현 정리	인터프리터
접근 잦은 객체는 동일 인터페이스 사용	반복자
객체 사이 상호작용을 캡슐화	중재자
객체 정보 저장이 필요할때 사용. 특정시점에서 내부 상태를 객체화해 객체를 특정 시점으로 돌리는 기능 제공. UNDO기능 개발시 사용	메멘토
객체의 상태가 변화하면 객체에 상속되어있는 곳에 상태를 전달하는 패턴	옵서버
알고리즘 개별 생성해 원할때마다 변경하는 패턴	전략
상위에선 골격, 하위에선 처리하는 패턴.	템플릿 메소드
처리기능 분리해 별도로 구성	방문자

<공통 모듈 테스트>

<정적 테스트> vs <동적 테스트>

1. 정적 테스트 (정형 명세 기법) -> 코드 실행 하지 않고 테스트
1. 워크스루 : 회의전 사전검토 후 회의진행
2. 인스펙션 : 다른 전문가가 검사
3. 동료검사 : 명세서 리뷰후 결함 발견
2. 동적 테스트
1. 화이트박스, 블랙박스

<화이트박스> - 응용프로그램 내부구조와 동작을 검사하는 소프트웨어 테스트

화이트 박스 테스트	
테스트 케이스 설계자가 논리적 복잡성을 측정할 수 있게 함	기초경로 검사
논리적인 조건 초점	조건 검사
반복구조 초점	루프 검사
변수정의나 위치 초점	데이터 흐름
<화이트박스 검증 기준>	
소스 코드 모든 부분을 한번이상 수행	문장 검증 (statement)
조건식에 상관없이 개별 조건식이 참 / 거짓 인 경우 한번 이상 수행되도록 구성하는 검증	조건(condition) 검증
모든 조건문에 대해 조건식이 참/거짓인 경우가 한 번 이상 수행 되도록 구성하는 검증	분기 검증

<화이트박스 검증 기준>		검증
모두 만족하는 조건검증으로, 조건문이 true false인 경우에 따라 조건 검증 기준의 입력 데이터를 구분하는 검증 기법		분기 / 조건 검증
결정조건 내 모든 개별 조건식의 모든 가능한 조합을 100% 보장하는 검증기법		다중 조건 검증
수행 가능한 모든 경로를 테스트, 맥케이브 순환복잡도 사용		기본 경로 검증
<블랙박스>		
과거 경험이나 감각 이용		오류-예측
정상/비정상 동작의 예상 결과를 동일한 수 만큼 테스트		동치분할 Equivalence partitioning
입력값의 경계에 있는 값으로 테스트		경계값 분석 (Boundary value)
다른 버전의 프로그램에 동일한 값을 넣어 동일한 결과가 나오는지 테스트		비교 (comparison)
분석 후 최적의 데이터로 테스트		원인-효과
<테스트 목적에 따른 분류>		
실패 유도후 정상적 복귀 여부	회복 테스트(Recovery Testing)	
보안적 결함을 미리 점검	안전 테스트(Security Testing)	
응답하는 시간, 업무량, 반응속도 측정	성능 테스트(Performance Testing)	
내부 논리 경로, 소스 코드의 복잡도를 평가	구조 테스트(Structure Testing)	
오류를 제거하거나 수정한곳에 새로운 오류 확인	회귀 테스트(Regression Testing)	
변경된 시스템과 동일한 데이터를 입력후 결과 비교	병행 테스트(Parallel Testing)	
<단위 모듈 구현 원리>		
변경 가능성 있는 모듈을 다른 모듈로부터 은폐	정보은닉	
복잡한 문제를 분해하고 모듈 단위로 문제해결	분할과 정복	
자료구조 액세스하고 함수 내에 자료구조 표현 내역 은폐	데이터 추상화	
낮은결합도와 높은 응집도	모듈 독립성	
<단위 모듈 재사용성>		
기존 기능 개선 또는 재활용	재공학	
sw에 대한 디버깅 같은 분석통해 알고리즘을 역으로 분석해 재구성	역공학	
기존 시스템 참조하여 새로운 시스템 개발	재개발	

<데이터 모델링>

데이터 모델링	특징	스키마	내용
개념적 모델링	객체간 관계 표현한 ERD 도출	개념적	사용자 요구에 대한 트랜잭션을 모델링하는 단계. 트랜잭션 모델링, 뷰 통합방법 등을 고려
논리적 모델링	관계, 키, 속성 도출	논리적	시스템에서 구현 가능한 테이블 구조로 변환. DBMS에 맞는 논리적 스키마 설계. 스키마 평가 수행.
물리적 모델링	저장구조와 테이블 정의서 및 명세서 산출	물리적	논리적 표현된 데이터를 DBMS같은 디스크 등의 하드웨어 저장장치에 저장할 수 있도록 변경.

<럼바우 분석기법 >

럼바우 분석기법	다이아그램	모델링
시스템에서 요구되는 객체 찾아 속성과 객체간 관계 규정	ERD	객체 모델링
시간 흐름에 따른 객체간 동적인 행위 표현	STD (상태변화도), 사건 추적도	동적 모델링
다수 프로세스들 간의 자료 흐름을 중심으로 처리과정 표현	DFD (자료흐름도)	기능 모델링

<개발방법론>

- SW 생명주기
 - 폭포수 모델
 - 프로토 타입 모델
 - 나선형 모델 (계획 및 정의 -> 위험분석 -> 개발 -> 고객평가)
 - 반복형 모델 (애자일)

개발방법론	방법론
개발에 필요한 관리절차와 작업기법을 체계화해 대형 프로젝트에 적합한 방법론	정보 공학 개발 방법론
특정 제품에 적용하고 싶은 공통기능을 정의해 임베디드 SW작성에 유용한 방법론	제품 라인 개발 방법론
애자일 유형으로, 수시로 발생하는 고객 요구 사항에 유연하게 대응하기 위해 개발과정 반복 (의 용 단 피 존) -> 의사소통, 용기, 단순성, 피드백, 존중	XP
력비용어에서 파생된 것으로, 매일 정해진 시간, 장소에서 짧은 개발하는 방법론 (백로그, 스프린트, 번 다운 차트)	스크럼
도요타의 린 시스템 품질기법으로 낭비요소 제거하여 품질 향상하는 방법론	린

<응집도 및 결합도>

<응집도> -(Cohesion)

서로 관련 없는 요소로 구성	우연적 (Concidental)
유사 성격이나 특정 형태로 된 요소로 구성	논리적 (Logical)
특정 시간에 처리되어야 하는 요소로 구성	시간적 (Temporal)
다수의 관련 기능을 가질 때 그 기능을 순차적으로 수행	절차적 (Procedual)
동일한 입출력으로 다른 기능 수행	교환적 (Communication)
내부 결과물을 다음 활동의 입력값으로 사용	순차적 (Sequential)
모든 기능이 단일 목적 위해 수행	기능적 (Function)

<결합도>

다른 모듈의 기능 및 자료를 직접 사용	내용
공유 데이터 영역을 여러 모듈이 사용	공통
어떤 모듈의 변수를 외부 모듈이 사용	외부
다른 모듈 내부의 논리 흐름을 제어하기 위한 제어 요소 전달 (권리전도현상 발생)	제어
모듈간 인터페이스 배열, 레코드의 자료구조 전달	스탬프
모듈간 인터페이스가 자료 요소만으로 구성	자료

<트랜잭션 특징>

<트랜잭션 특징>

트랜잭션의 연산은 DB에 모두 반영되도록 완료(Commit) 되거나 반영되지 않도록 복구 (Rollback) 되어야 한다.	원자성 (Atomicity)
트랜잭션이 실행을 성공적으로 완료하면 언제나 일관성있는 데이터베이스 상태로 변환 함	일관성 (consistency)
트랜잭션 병행 실행되는 경우, 하나의 트랜잭션 실행중에 다른 트랜잭션이 연산에 끼어 들 수 없음.	독립성 (Isolation)
성공적으로 완료된 트랜잭션은 시스템이 고장나더라도 영구적으로 반영	영속성 (Durability)

<트랜잭션 병행제어>

- 동시에 여러개의 트랜잭션을 병행 수행할 때, 트랜잭션들이 DB의 일관성을 파괴하지 않도록 트랜잭션 간의 상호작용을 제어하는 것

<트랜잭션 병행제어>

트랜잭션 수행할 때 lock의 허락이 있어야만 수행	로킹
------------------------------	----

<트랜잭션 병행제어>

트랜잭션 간 수행 시간표를 정해 작업 수행	타임 스탬프
트랜잭션이 어떠한 검증도 하지 않고 수행하고, 종료시 검증 수행해 DB에 반영	낙관적 검증
타임 스탬프 이용한 기법	MVCC(다중버전기법)

<트랜잭션 병행제어 미보장>

- 트랜잭션 병행제어 미보장시 발생유형

<트랜잭션 병행제어 미보장>

먼저 실행된 트랜잭션 결과를 나중 실행된 트랜잭션 결과로 덮어 씌워진 경우	갱신손실
트랜잭션 중간 수행 결과를 다른 트랜잭션이 참조	현황 파악 오류
두 트랜잭션 동시 실행되 DB일관성에 문제 생김	모순성
복수 트랜잭션 데이터 공유시 특정 트랜잭션 취소할때 부분취소 안되는 경우	연쇄복귀

<트랜잭션 회복기법>

- 트랜잭션들을 수행하는 도중 장애가 발생해 DB가 손상되었을 때 손상되기 전의 정상 상태로 복구하는 작업

<트랜잭션 회복기법>

트랜잭션 처리 완료될때 까지 DB에 대한 실질적인 갱신을 연기	지연갱신 기법 (Deffered update)
트랜잭션이 데이터 갱신하면 부분 완료되기 전이라도 즉시 실제 DB에 반영. REDO와 UNDO모두 사용가능	즉각갱신 기법 (immediate update)
트랜잭션 실행시 특정 단계에 재실행 할 수 있도록 검사점을 로그에 보관하여 장애 발생시 모두 철회하지 않고 해당 부분만 철회가능	검사점 기법 (check point)
갱신 이전의 DB를 일정 크기의 페이지 단위로 나눠 복사된 그림자 페이지 별도로 저장	그림자 페이지 대체 기법 (shadow paging)

<트랜잭션 명령어>

<트랜잭션 명령어>

트랜잭션 롤백해 이전 상태로 데이터베이스 되돌림	UNDO
UNDO반대작업으로, UNDO된 트랜잭션을 다시 데이터베이스에 적용	REDO
데이터베이스 무결성 검사하고 확인하는 작업	check
트랜잭션 변경 사항 취소 후 이전상태로 되돌림	ROLLBACK
트랜잭션 변경사항 확정 후 저장	COMMIT

<트랜잭션 명령어>

트랜잭션중간 저장점 설정	SAVEPOINT
---------------	-----------

<DB 보안 요소>

<DB 보안 요소>

시스템의 자원과 정보는 인가된 사용자에게만 접근이 허락된다.	기밀성 (Confidentialit)
시스템 자원은 오직 인가된 사용자만 사용하며 수정 및 변경할 수 있다.	무결성 (Integrity)
인가받은 사람은 언제 어디서든 사용할 수 있다.	가용성 (Availability)
사용자의 신원을 인정	인증(authentication)
인증된 주체에게 접근권한 허용	인가(Authorization)
데이터 송수신 증거 제시하는 방법	부인방지

<소스코드 도구>

정적 분석 도구 (pmd, Cppcheck 빼고 나머지가 크로스 플랫폼 지원)

미사용 변수, 최적화 안된 코드 ,결함 유발 코드 검사 (리눅스 , 윈도우)	Pmd
C와 C++ 코드에 대한 메모리 누수, 오버플로우 검사 (윈도우)	Cppcheck
중복코드 , 복잡도, 코딩설계등을 분석하는 소스 분석 통합 플랫폼	SonarQube
자바 코드에 대해 코드 표준 검사	Checkstyle
다양한 언어의 코드 복잡도 분석	Ccm
자바 언어의 소스 코드 복잡도 분석	Cobertura

동적 분석 도구

Valgrind 프레임워크 및 STP 기반으로 구현	Avalanche
프로그램 내에 존재하는 메모리 및 쓰레드 결함 분석	Valgrind

< 빌드도구>

< 빌드도구>

Ant	
Make	
Maven	
groovy 기반으로 한 안드로이드 앱 개발환경에서 사용되는 도구. 명령어 일들을 모아 태스크 단위로 처리	Gradle
java기반 서블릿 컨테이너에서 실행되는 도구. 분산빌드나 테스트 기능 지원	Jenkins

<인터페이스 구현 검증 도구>

<인터페이스 구현 검증 도구>

Java, C++ 등 다양한 언어 지원하는 단위테스트 프레임워크	XUnit
컴포넌트 재사용 등 다양한 환경 제공 프레임워크	STAF
웹 기반 테스트 프레임워크	FitNesse
FitNesse의 협업기능과 STAF를 통합한 NHN프레임워크	NTAF
다양한 브라우저 지원하는 웹 어플리케이션 테스트 프레임워크	Selenium
Ruby사용	Watir
Db에이전트를 통해 애플리케이션 모니터링	스카우터
애플리케이션 운영,안정화 전까지 모든 생명주기 성능 모니터링	제니퍼

<페이지 교체 알고리즘>

<페이지 교체 알고리즘>

가장 오랫동안 사용하지 않을 데이터 삭제	OPT
먼저 들어온 것부터 삭제	FIFO
가장 오랫동안 사용되지 않은거부터 삭제	LRU
참조 횟수가 가장 적은거부터 삭제	LFU
참조 횟수가 가장 많은것부터 삭제	MFU
최근 사용하지 않은 것부터 삭제	NUR

<프로세스 스케줄링>

선점형

시분할 시스템 이용해 같은 CPU시간 할당	라운드로빈
짧은 수행시간 프로세스 우선 수행	SRT
독립된 스케줄링 큐	다단계 큐
큐마다 다른 시간 할당	다단계 피드백 큐

비선점형

동일한 우선순위면 FCFS로 사용	우선순위
큐 도착한 순서대로 CPU할당	FCFS
요청에 명세된 시간 내 처리	기한부
단기 작업 우선으로, 짧은 실행시간 문저 CPU할당	SJF

비선점형

SJF 보완한 것으로, 우선순위 계산법 이용한 응답률(대기+실행)/실행 공식 이용해 계산 HRN

<암호화 알고리즘>

1. 블록 암호화
1. 한번에 하나의 데이터 블록을 암호화 하는 방법
2. 스트림 암호화
1. 평문과 같은 길이의 스트림을 생성해 비트 단위로 암호화 하는 방법

<암호화 알고리즘>	연도	알고리즘	개발한 곳	내용
Skipjack		블록	NSA	클리퍼칩 이용한 음성통신에 사용됨
DES	1975	블록	NBS	키 길이가 56bit 으로 짧음
IDEA		블록	스위스 연방기술 기관	DES 대체. 키길이가 128Bit 고정
CBC	1976	블록	IBM	각 블록은 암호화 되기 전에 이전 블록 결과와 XOR 함
SEED	1999	블록	한국인터넷진흥원	
AES	2001	블록	NIST	DES 발전. 키 길이가 유연함
ARIA	2004	블록	국가정보원과 산악연합회	
LFSR		스트림		레지스터 입력값이 이전 상태 값들의 선형함수로 계산
RC4		스트림		
디피헬만		비대칭키		최초 공개키 암호화 방식. 이산대수의 계산 어려운 문제 기본원리로 함
RSA		비대칭키		소인수분해 어려운거 이용한 방법
ECC		비대칭키		RSA보완한 타원곡선 함수
Hash				임의의 길이를 입력받으면 고정된 길이 값을 출력
SHA			미국 국가안보국	미국 국가 표준으로 지정한 해시 암호화 알고리즘

<RAID 계층>

- 스트라이핑 (0~3번)
- 여러 디스크에 데이터를 분산 저장하는 것

- 복제하지 않으면 하나의 디스크만 문제 있어도 전체 RAID에 문제 생김
- 패리티 디스크 (3~6번)
 - 패리티 디스크를 별도로 생산해 관리하는 방법
 - 백업전용 디스크로, 복구 가능하도록 앞의 디스크 정보들을 저장하고 있는 디스크이다.

<RAID 계층>	방식	내용
레이드 0번	스트라이핑	여러 디스크에 분산저장해 동시에 읽어들임. 중복저장x
레이드 1번	미러링 방식	미러링 방식 사용. 중복저장o
레이드 2번	스트라이핑	해밍코드 및 에러검증 사용
레이드 3번	스트라이핑	바이트 단위
레이드 4번	패리티	블록단위 패리티
레이드 5번	패리티	패리티 여러 하드에 분산
레이드 6번	패리티	패리티 이중구조 저장

<정규화>

정규화	특징	내용
1NF	완전 함수적 종속 제거	필드값이 원자값이 될 수있도록 함
2NF	부분 함수적 종속 제거	기본키가 복합키로 이루어졌을 경우, 복합키 중 하나가 필드의 결정자일때 복합키 유지 상태로 결정자인 키와 필드를 하나의 테이블로 분리
3NF	이행 함수적 종속 제거	A->B 이고, B->C 인경우 A->C 를 만족하도록 분리
BCNF	결정자 제거	보이스 코드 정규화로, 복합키를 결정자를 이용해 분리하는 것으로, 결정자가 후보 키가 되도록 테이블 분해
4NF	다중 종속성 제거	
5NF	조인 종속성 제거	

<교착상태 해결 방법>

- 발생 원인
 - 상호배제 : 한번에 한개만 자원 공유
 - 점유와 대기 : 하나 자원 점유하며 다른 자원 추가 점유시 대기하는 프로세스 존재
 - 환형대기 : 공유자원 사용하기 위해 대기하는 프로세스들이 원형 구조 이룸
 - 비선점 : 사용 끝날때까지 뺏지 못함

교착상태 해결 방법 - pard 로 외우기

교착상태 해결 방법 - pard 로 외우기

네가지 조건 중 하나 제거	예방 (Prevention)
다익스트라 은행원 알고리즘이 예시임. 교착상태 발생시 적절히 회피	회피 (Avoidance)
교착상태 일으킨 프로세스 종료시켜 회복	회복 (Recovery)
점검을 통해 교착상태 프로세스와 자원을 발견해 해결	발견 (Detection)

<OSI 7계층>

OSI 7계층	사용장치	데이터 형태	프로토콜
1. 물리	리피터, 허브	비트	RS-232
2. 데이터 링크	스위치	프레임	이더넷
3. 네트워크	라우터	패킷	IP, ICMP, ARP
4. 전송	게이트웨이	세그먼트	TCP, UDP
5. 세션		메시지	SSH
6. 표현		메시지	JPG, MPEG, PAP
7. 응용		메시지	HTTP, FTP, DNS, SMTP

<네트워크 보안 프로토콜>

- 패킷 교환방식
 - 가상회선 방식 : 단말기 간에 논리적 가상회선을 미리 설정해 연결 확립
 - 데이터그램 방식 : 연결경로 설정하지 않고 전송해 순서 없이 전송. IP에서 사용됨

네트워크 보안 프로토콜	내용
3계층에서 인증 보장하는 인증헤더와 무결성, 기밀성을 보장하는 암호화를 이용한 IP 보안 프로토콜	IPsec
IPSec의 무결성을 보장해 데이터 인증 제공	AH
IPSec의 기밀성을 보장해 암호화 알고리즘 활용한 캡슐화 기반 페이로드 제공	ESP
초기 무선 네트워크에서 사용되던 보안 프로토콜, 기밀성 제공위한 무선 통신에 사용	WEP
WEP 대체한 무선 Wi-Fi 보안에서 사용되는 암호화 프로토콜. RC4 알고리즘 기반	TKIP
전송 계층과 응용계층 사이에서 클라이언트와 서버 간 웹 데이터 암호화.상호인증 및 전송시 데이터 무결성 보장하는 프로토콜	SSL / TLS
웹 상에서 네트워크 트래픽을 암호화 하는 주요 방법중 하나로, 클라이언트와 서버간 전송되는 모든 메시지 암호화 하는 프로토콜	S-HTTP
인터넷에 연결된 서로 다른 기종의 컴퓨터들이 데이터를 주고 받을 수 있도록 하는 표준 프로토콜	TCP/IP
3계층에 해당하며 데이터그램 방식을 기반으로 한 비연결형 프로토콜	IP

네트워크 보안 프로토콜	내용
4계층에 해당하며 가상회선 방식을 기반으로 한 연결형 프로토콜 로 양방향 서비스를 지원한다.	TCP
4계층에 해당하며 데이터그램 방식을 사용하여 데이터 전송 전에 연결을 설정하지 않는 비연결형 프로토콜 로 단순헤더 구조로 인해 빠르다.	UDP
온라인상 안전한 거래를 위해 VISA와 Master Card에서 개발한 프로토콜	SET
공개키 기반 구조로, 인증기관에서 전자서명된 인증서 발급받아 안전하게 비밀통신하는 기술	PKI
전자 문서에 서명 했다는 사실을 나타내는 전자적 형태정보	전자 서명

<네트워크 용어>

<네트워크 용어>	
차세대 이동 통신, 홈네트워킹에 사용되는 대규모 디바이스 생성에 최적화 된 기술	메시 네트워크
운영체제의 프로세스간 서로 데이터 주고받기 위한 통신기술.(파이프, 네임드 파이프, 메시지큐, 공유 메모리, 소켓, 시그널)	IPC
근접에서 기가급 속도 전송 가능한 초고속 근접 무선 통신기술	Zing
여러개의 독립된 장치가 블루투스나 UWB 이용해 통신망 형성하는 것	피코넷
재난현장같은 유선연결이 어려운 환경에서 노드같은 모바일 호스트만을 이용해 통신망 구축	ad-hoc 네트워크
광섬유 이용한 통신기술로, 파장 다르게 해서 여러대의 단말기가 동시에 같은 회선에서 다른 내용의 통신을 할 수 있게 하는 것	WDM(파장 분할 다중화)
실세계와 가상세계의 다양한 사물들을 인터넷으로 연결해 진보된 서비스 제공	IOT
컴퓨팅 자원을 중앙에 두고 인터넷을 통해 언제 어디서든 작업 수행할 수 있는 가상화된 환경	클라우드 컴퓨팅
링크 데이터와 오픈 데이터의 합성어로, 누구나 사용할 수 있도록 웹상에 공개된 연계 데이터(웹상에 존재하는 데이터를 개별 URL로 식별)	LOD
데이터센터의 모든 자원을 가상화하여 인력 개입없이 소프트웨어 조작만으로 관리 및 제어되는 데이터 센터	SDDC
각종 센서로 수집한 정보를 무선으로 수집할 수 있도록 구성한 네트워크	USN
tcp의 세션관리 취약점을 이용해 victim과 server 사이 패킷 스니핑	TCP Session Hijacking
중심 주파수의 20% 이상 점유한 대역폭을 가지는 신호로, 초광대역 사용하는 초고속 무선 네트워크 전송기술	UWB
도난당한 스마트폰의 작동을 웹사이트를 통해 정지할 수 있는 기술	킬스위치
유선 랜 반이중 방식으로, 현재 채널 사용중인지 체크하여 전송하는 MAC 방식	CSMA/CD

<네트워크 용어>	
무선 랜 반이중 방식으로, 사전에 가능한 충돌을 회피하는 무선 전송 다원 접속 방식	CSMA/CA
에너지 이용 효율을 극대화 하는 전력망으로, 전력망을 지능화 함으로써 고품질 전력 서비스 제공	스마트 그리드
근거리 무선 통신 기술로, 스마트 그리드와 연계하여 에너지 효율적으로 관리할 수 있도록 특화된 무선 통신 기술	Wi-Sun
네트워크 주소 변환으로, 1개의 정식 IP 주소에 여러 가상사설IP 주소를 할당 및 연결하는 방식	NAT
인터넷 제어 메시지 프로토콜로, IP 패킷 처리시 발생하는 오류의 처리와 전송 경로 변경을 위한 제어 메시지를 관리 하는 역할	ICMP
호스트 컴퓨터와 인접 라우터가 멀티캐스트 그룹 멤버쉽 구성해 사용	IGMP
http, smtp 등을 사용하여 xml 기반의 메시지를 네트워크 상에서 교환하는 프로토콜	SOAP
속성 값이 키-값으로 되어있어 빅동기 처리인 Ajax에서 xml 대신 사용됨	JSON
특수한 목적을 갖는 마크업 언어를 만드는데 사용되는 다목적 마크업 언어	XML
저속 전송속도로 홈 네트워크 위한 표준, 메시 네트워크 기반으로 동작	Zigbee
CHIP프로젝트에서 IP 기반으로 스마트홈의 표준 제작. thread 채택해 사용중임.	Matter
웹 서비스에 대한 상세 정보가 XML 형식으로 기술되어 있는 언어	WSDL
저장소로, WSDL을 등록하고 검색하기 위한 저장소.	UDDI
P2P기반으로 문서를 분산저장하는 기술	분산 원장 기술
P2P 네트워크 이용해 온라인 금융거래 정보를 참여자 Peer에게 분산저장하는 것	블록체인

<정보보안 용어 정리>

정보보안 용어 정리	내용
헤더 최종끝 알리는 개행문자 이용해 대상 웹 서버와 장시간 지속 연결해 자원 모두 소진시키는 공격기법	Slowloris
tcp 윈도우 크기를 낮게 설정해 서버로 전달하고 데이터 전송 완료시까지 연결 유지해 자원고갈	Slow read attack
사전에 사이트 감염시켜 사이트 방문시 악성 코드 감염되게 함	워터링 홀
자신 복제해 시스템에 부하 높임, DDos, 버퍼 오버플로, 슬래머가 속함	웜
보안 취약점 공표전 신속하게 공격	제로데이
사용자 입력 탐지해 공격	키로거
내부 문서 암호화해 금전적 요구함	랜섬웨어
시스템 설계자가 만든 구멍을 이용	백도어

정보보안 용어 정리	내용
정상인것처럼 잠입해 특정 동작시 실행, 웜과 차이점은 자기복제 안한다는 것임	트로이목마
SMS(문자) 이용해 개인정보 탈취	스미싱
네트워크 중간에서 남의 패킷 정보를 도청하는 해킹유형	스니핑
이메일 통해 링크나 첨부파일 클릭하면 정보탈취하는 공격유형	스피어 피싱
사이트 접속시 철자 실수한걸 이용해 유사 사이트로 이동시키는 공격유형	타이포스 퀴팅
IP나 ICMP 이용해 데이터 집중적으로 보내 네트워크 불능 상태로 만드는 공격	스머핑
ARP 취약점 이용해 자신의 Mac 주소를 공격대상의 Mac주소로 변경해 패킷 가로채는 공격	ARP 스푸핑
IP패킷 재조합 과정에서 잘못된 세크먼트 오프셋 정보로 인해 수신 시스템에 문제 발생시켜 공격	티어 드롭
패킷 재전송과 재조립에 과부하 발생시키는 공격방법으로, 같은 시퀀스 번호 계속 보냄	붕크
일정 간격으로 시퀀스 번호에 빈 공간 생성하는 공격 방법	보잉크
TCP 구조적 문제 이용해 ACK이 아닌 SYN 패킷만 보내 자원 고갈시키는 공격방법	SYN 플러딩
대량의 UDP 패킷을 만들어 임의의 포트로 전송 후 응답 메시지 생성하게 만들어 자원 고갈	UDP플러딩
ICMP 패킷 정상적인 크기보다 아주 크게 만들어 IP 단편화 발생시키는 공격방법	죽음의 핑
출발지 IP와 목적지 IP를 같은 패킷 주소로 보내 수신자가 자기 자신에게 응답보내게 하는 공격방법	랜드 어택
메시지를 공격자가 원하는 형태로 만들어 특정 목적지로 가는 패킷 탈취	Icmp Redirection
사용자 의지없이 특정 웹사이트로 공격자 의도대로 요청하는 것	CSRF
부적절한 웹페이지 열람해 부적절한 스크립트 실행	XSS
입력칸에 SQL 구문 삽입해 DB 접근하여 정보 탈취	SQL Injection
하드웨어 및 소프트웨어 구성의 취약점 파악을 위해 공격자가 취약점 탐색	네트워크 스캐너/스니퍼
인증된 호스트의 IP 어드레스로 위조해 타깃에 전송하는 공격기법	IP스푸핑
특정 타깃을 목표로해 다양한 수단을 통한 지속적이고 지증적인 맞춤형 공격기법	APT
소프트웨어 개발사의 서버에 접근해 업데이트시 감염시키는 공격기법	공급망 공격
무선 WIFI피싱 기법으로 핫스팟에 연결한 사용자들의 정보 탈취	이블트윈 공격
평범한 SW인지 구분하기 어려운 중간에 위치하는것을 총칭하는 언어	그레이웨어
결제자의 다양한 정보 수집해 패턴 만든 후 패턴과 다른 이상결제 잡아내는 보안 시스템. 빅데이터 바탕으로 구축됨	FDS(이상 행위 탐지 시스템)

<네트워크 도구>

네트워크 도구	내용
UDP Flood 서비스 거부 공격 유발에 사용되며 몇개의 서버(마스터)와 많은 클라이언트(데몬)으로 이루어진 공격도구	Trinoo
많은 소스에서 여러개의 목표 시스템에 대해 서비스 거부 공격 수행할 수 있는 공격도구	Tribe Flood Network
분산 서비스 거부 에이전트 역할을 하는 Linux 및 Solaris 시스템용 공격도구	Stacheldraht
시스템 침입 후 사실 숨긴채 차후 해킹할 수 있도록 프로그램 설치하는 공격 도구. 로그도 지울수 있음	루트킷
공격용 툴킷으로, 온라인 상에서 불법적인 행위를 수행하기 위한 것으로, pc에 설치되 정 보수집합	크라임웨어
크래커 침입해 시스템에 백도어 만들어 놓거나 설정파일 변경시 알려주는 도구	Tripwire
패킷의 내용을 출력해주는 스니핑 도구의 일종으로 도청에 대한 추적 및 감지하는 도구	Tcpdump

<네트워크 장비>

네트워크 장비	내용
외부 네트워크와 접속해 가장 빠른 속도로 데이터 주고받을 수 있게 컴퓨터 내에 설치되는 장치	NIC (Network Interface Card)
현재 위치한 네트워크에서 다른 네트워크로 데이터 보내거나 다른 네트워크로부터 데이터 받는 출입구역화하는 장치	게이트웨이

<DDOS 분산 서비스 거부 공격 >

DDOS 분산 서비스 거부 공격	내용
공격하는 사람	공격자
공격자의 명령을 받아 에이전트를 컨트롤하는 서버	마스터(핸들러)
공격대상 서버를 명령을 받아 공격하는 곳	에이전트(데몬)
악성코드 감염된 PC로, C&C 서버 제어받아 DDos에 이용됨	зомбиPC
해커가 зомбиPC에게 명령내려 악성코드 제어하는 서버	C&C
감염된 зомбиPC로 이루어진 네트워크	ботнет

<클라우드 및 가상화>

클라우드 및 가상화	내용
리눅스 재단에 의해 관리되는 컨테이너화된 애플리케이션을 자동배포 및 관리하는 오케스트레이션 플랫폼	쿠버네티스

클라우드 및 가상화	내용
리눅스의 응용 프로그램들을 프로세스 격리 기술 사용해 컨테이너로 실행하고 관리하는 SW	도커