

07-기-1 ❖ **데이터베이스 정의**

07-기-2

07-산-1

07-산-2

- 특정 조직이 업무 수행하는 데 필요한 관련성 있는 자료들의 집합체
- (통합(integrated data), 저장(stored data), 운영(operational data), 공용(shared data))
- 동일 데이터의 중복성을 최소화해야 한다.
- 컴퓨터가 접근할 수 있는 저장 매체에 저장된 자료이다.
- 조직의 존재 목적이나 유용성 면에서 존재 가치가 확실한 필수적 데이터이다.

07-기-2 ❖ **데이터베이스 특징**

07-기-3

07-산-2

- 실시간 접근(Real-time accessibility) : 내가 원할 때마다 언제든지 바로 접근해서 자료를 처리할 수 있다.
- 계속적인 변화(Continuous evolution) : 데이터의 삽입, 삭제, 갱신 작업으로 항상 최신의 데이터를 유지해야 한다.
- 공용(Concurrent sharing) : 여러 사용자가 같이 쓸 수 있어야 한다.
- 내용에 의한 참조(Content reference) : 사용자가 요구하는 데이터 내용으로 데이터를 찾는다.

07-산-3 ❖ **정보와 데이터의 정의**

- 정보 : 데이터를 가공 처리하여 얻은 지식, 유용한 정보
- 데이터 : 숫자, 문자, 기호로 구성된 실질적인 자료

07-산-3 ❖ **DBMS 필수 기능**

- 정의 : 데이터베이스 자료형, 구조, 이용방법, 제약조건을 명시
- 조작 : 검색, 저장, 삭제, 갱신 기능
- 제어 : 데이터의 정확성, 안정성 유지

07-기-3 ❖ **스키마 특징**

- 데이터 사전(= 시스템 카탈로그)에 저장
- 데이터베이스에 저장되어 있는 모든 데이터 개체들에 대한 정보를 유지, 관리하는 시스템
- 데이터베이스의 구조(개체, 속성, 관계)에 대한 정의
- 다른 이름으로 메타데이터(데이터의 데이터)라고 함

07-기-3 ❖ **스키마 3계층**

07-산-1

07-산-3

- 외부 스키마 (= 서브 스키마 = 사용자 뷰)
 - 사용자가 보는 관점 (사용자에 따라 다름, 여러 개 존재)
- 개념 스키마 (= 스키마 = 전체적인 뷰)
 - DB 전체적인 논리적 구조
 - 개체간의 관계와 제약 조건을 나타내고 데이터베이스의 접근 권한, 보안 및 무결성 규칙을 명세화 한다
- 내부 스키마 (실제 Data를 저장)
 - DB 전체적인 물리적 구조
 - DBA 관리

07-산-2 ❖ **DB 관리자 (DBA : DataBase Administrator)**

- 데이터베이스 설계와 조작에 대한 책임
- 행정적 책임
- 시스템 감시 및 성능 분석

07-기-1 ❖ **데이터베이스 설계 순서**

07-기-3 -요구조건 분석->개념적 설계->논리적 설계->물리적 설계->구현

07-산-1

07-산-2

07-산-1 ❖ **개념적 설계**

07-산-2 -개체 타입과 이들 간의 관계 타입을 이용해 현실 세계를 개념적으로 표현
-(E-R 다이어그램)

07-기-1 ❖ **논리적 설계**

-목표 DBMS에 맞추어 논리적 모델로 설계 (관계형, 계층형, 망형 모델)
-트랜잭션 인터페이스 설계, 스키마의 평가 및 정제

07-기-3 ❖ **물리적 설계**

-물리적 구조 데이터 표현
-저장 레코드 양식 설계
-레코드 집중의 분석 및 설계
-접근 경로 설계

-예문>데이터베이스의 물리적 설계 옵션 선택시 고려사항으로 거리가 먼 것은?
-스키마의 평가
-응답시간
-저장 공간의 효율화
-트랜잭션 처리도(throughput)

07-기-3 ❖ **데이터 모델 구성요소**

-구조(structure) : 개체들 간의 관계
-연산(operation) : 데이터 처리하는 방법
-제약조건(constraint) : 실제 데이터의 논리적인 제약조건

07-산-3 ❖ **Null의 의미**

07-기-2

-예문>데이터베이스에서 널(Null) 값에 대한 설명으로 옳지 않은 것은?
-아직 모르는 값 또는 알려지지 않은 값을 의미한다.
-공백이나 0(Zero)과 같은 의미이다
-정보 부재를 나타내기 위해 사용한다.

07-기-1 ❖ **ER (Entity-Relationship, 개체관계도) 모델**

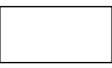
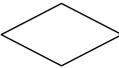
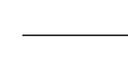
07-산-2

07-기-2

07-산-1

07-산-2

07-산-3

| 의미 | 개체 | 관계 | 속성 | 기본키 속성 | 연결 |
|----|---|---|--|---|---|
| 기호 |  |  |  |  |  |

❖ **논리적 데이터 모델의 종류**

-관계형 : Table구조
-계층형 : Tree구조
07-산-3 -네트워크형 : Graph구조
-사이클이 허용되지 않고, 개체 삭제 시 연쇄 삭제 발생,
-구조 간단 -> 구현,수정,검색 쉽다. / 삽입, 삭제가 어렵다.

❖ 관계형 데이터베이스의 릴레이션 구조

| 학번 | 이름 | 주소 | 성별 |
|------|-----|-----------|----|
| A001 | 조대호 | 대구시 | 남 |
| A002 | 장동건 | 서울시 | 남 |
| A003 | 김태희 | 부산시 | 여 |

- 07-산-1 -차수(Degree): 속성들의 수
- 카디날리티(cardinality): 튜플들의 수

07-기-1 ❖ 릴레이션 특징

- 07-기-3 -한 릴레이션에 정의된 튜플들은 모두 다르다.
- 07-산-2 -한 릴레이션에 정의된 튜플들은 순서에 무관하다.
- 07-산-3 -튜플들은 시간에 따라 변한다.
- 07-산-3 -릴레이션 스키마를 구성하는 속성들도 순서에 무관하다.
- 속성의 명칭(이름)은 유일해야 하지만, 속성의 값은 동일해도 된다.
- 속성은 더 이상 쪼갤 수 없는 원자값으로 구성된다.
- 릴레이션을 구성하는 튜플을 유일하게 식별하기 위한 속성들의 부분집합을 키(Key)로 설정한다.

❖ 키(Key)

- 슈퍼키(Super key) : 한 릴레이션 내에 있는 속성들의 집합으로 구성된 키를 말한다. (유일성)
- 후보키(Candidate key) : 한 릴레이션 내에 있는 모든 튜플들을 유일하여 식별할 수 있는 하나 또는 몇 개의 애트리뷰트 집합(최소 슈퍼키 : 유일성 + 최소성)
- 기본키(Primary Key) : 후보키 중에 선택한 키 (중복되어서는 안되며, Null 값을 가질 수 없다.)
- 대체키(Alternate key) : 후보키 중에서 기본키를 제외한 속성들
- 07-기-2 -외래키(Foreign key) : 어떤 r에서 다른 r을 참조할 때 참조 기준이 되는 속성으로서 참조하고자 하는 R의 기본키와 동일



07-기-2 ❖ 무결성

- 07-기-3 -참조 무결성 : 릴레이션은 참조할 수 없는 외래키값을 가질 수 없음을 의미하는 제약 조건
- 07-산-2 -개체 무결성 : 한 릴레이션의 기본키를 구성하는 어떠한 속성 값도 널(NULL) 값이나 중복 값을 가질 수 없다
- 07-산-3 -도메인 무결성 : 각 속성 값은 반드시 정의된 도메인에 속한 값이어야 한다.

❖ **관계대수와 관계해석의 정의**

- 관계대수 : 질의들을 대수로 표현하며 대수로 표현된 질의들은 다양한 연산자를 사용해서 구성한다
- 관계해석 : 집합으로 기출하며 해답이 계산되는 방법을 명시하지 않고 원하는 답을 기술하는 방식(비절차적, 선언적)

07-산-3

- 예문> 관계대수와 관계해석에 대한 설명으로 옳지 않은 것은?
 - 기본적으로 관계대수와 관계해석은 관계 데이터베이스를 처리하는 기능과 능력면에서 동등하다.
 - 관계해석에는 튜플 관계해석과 도메인 관계해석이 있다.
 - 관계해석은 수학의 프레디컷 해석(Predicate Calculus)에 기반을 두고 있다.

❖ **SELECT (형식 : σ 조건 (R))**

- 릴레이션에서 주어진 조건을 만족하는 튜플들을 검색하는 것으로 기호는 그리스 문자의 시그마(σ)를 이용한다.(수평적 연산)

❖ **PROJECT (형식 : π 속성 (R))**

- 릴레이션에서 주어진 조건을 만족하는 속성들을 검색하는 것으로, 기호는 그리스 문자의 파이(π)를 이용한다. (수직적 연산)

- 예문>

07-기-1

- π 이름, 학년 (σ 학과 + '컴퓨터' (학생))
 - SELECT 이름, 학년 FROM 학생 WHERE 학과 = '컴퓨터';

07-기-2

- π name, dept(δ year =3 (student))
 - SELECT name, dept FROM student WHERE year =3 ;

07-산-1

- π 이름 (σ 학과='컴퓨터'(학생)) → 컴퓨터 학과 학생의 이름을 검색하라.

❖ **JOIN**

- 두 개의 릴레이션 A와 B에서 공통된 속성을 연결하는 것이다.

07-기-2

❖ **DIVISION(\div)**

- 나누어지는 릴레이션인 A는 릴레이션 B의 모든 내용을 포함한 것이 결과 릴레이션이 된다

❖ **합집합 (U)**

- 릴레이션 A 또는 B에 속하는 튜플들로 구성된 릴레이션이다

❖ **교집합 (\cap)**

- 릴레이션 A 와 B에 공통적으로 속하는 튜플들로 구성된 릴레이션이다.

❖ **차집합 ($-$)**

- 릴레이션 A에만 있고 B에는 없는 튜플들로 구성된 릴레이션이다

❖ **카티션 프로덕트(cartesian product) (X)**

- A에 속한 각 튜플 a에 대하여 B에 속한 튜플 b를 모두 접속시킨 튜플들(a b)로 구성된 릴레이션이다.

07-기-2 ❖ 정규화 (normalization)

-정규화를 하는 이유는 데이터의 중복을 방지하고 보다 효율적으로 데이터를 저장하기 위함.(릴레이션 분리 -> 삽입, 삭제, 갱신 이상의 발생 가능성을 줄이는 것)

07-기-1 ❖ 이상(Amomaly)

07-기-2

-릴레이션에서 일부 속성들의 종속으로 인해 데이터의 중복이 발생하여 테이블 조작 시 불일치가 발생하는 것

07-산-1

- 갱신 이상 : 반복된 데이터 중에 일부만 수정하면 데이터의 불일치가 발생
- 삽입 이상 : 불필요한 정보를 함께 저장하지 않고는 어떤 정보를 저장하는 것이 불가능
- 삭제 이상 : 유용한 정보를 함께 삭제하지 않고는 어떤 정보를 삭제하는 것이 불가능

❖ 정규화과정

07-산-2

-제1정규형(1NF) : 반복 되는 속성을 제거한 뒤 모든 속성이 원자 도메인 만으로 되어 있는 정규형

07-산-3

-제2정규형(2NF) : 부분함수적 종속을 제거하여 완전 함수적 종속을 만족하는 정규형.

07-기-1

-제3정규형(3NF) : 이행적 함수적 종속 관계 제거하여 비이행적 함수적 종속 관계를 만족하는 정규형

07-기-2

-BCNF(Boyce/Codd Normal Form) : 결정자가 후보키가 아닌 함수 종속 제거 모든 결정자가 후보키이어야 한다는 것

-제4정규형(4NF) : 다치 종속 제거

-제5정규형(5NF) : 조인 종속성 이용

- 07-산-2 ❖ **시스템 카탈로그**
- 07-기-3 - 시스템 자신이 필요로 하는 여러 가지 객체에 관한 정보를 포함하고 있는 시스템 데이터베이스
- 특징
 - 데이터베이스 시스템에 따라 상이한 구조를 가진다
 - 사용자도 SQL을 이용하여 검색할 수 있다.
 - 객체들로서는 기본 테이블, 뷰, 인덱스, 데이터베이스, 패키지, 접근 권한 등이 있다.

- 07-산-1 ❖ **SQL 구분**
- 07-기-3 - 정의어: DDL (CREATE, ALTER, DROP)
정의 변경 제거
- 07-산-1 - **도메인, 테이블, 뷰, 인덱스를 정의, 변경, 제거하는 언어**
- 07-산-3 - 논리적 데이터 구조와 물리적 데이터 구조의 정의
- 07-산-2 - 논리적 데이터 구조와 물리적 데이터 구조 간의 사상 정의
- 07-기-1 - 번역한 결과가 데이터 사전에 저장
- 조작어: DML (SELECT, INSERT, DELETE, UPDATE)
검색 삽입 삭제 갱신
- 제어어: DCL (GRANT, REVOKE, COMMIT, ROLLBACK)
권한부여 권한취소 transaction 제어

- 07-산-1 ❖ **테이블을 생성 합니다.**
`CREATE TABLE` 학과 (학과코드 CHAR(18), 학과명 CHAR(18));
- ❖ **자료(튜플)를 입력 합니다.**
`INSERT INTO` 학과 `VALUES` ('A001','정보');
- 07-산-2 ❖ **자료(튜플)를 검색 합니다**
`SELECT * FROM` 학과;
- 07-기-1 ❖ **자료(튜플)를 수정 합니다.**
`UPDATE` 학과 `SET` 학과명='사무' `WHERE` 학과코드='A001';
- 07-기-3 ❖ **자료(튜플)를 삭제 합니다.**
`DELETE FROM` 학과;
- ❖ **테이블을 삭제 합니다.**
`DROP TABLE` 학과;

❖VIEW 정의

-사용자에게 접근이 허용된 자료만을 제한적으로 보여주기 위해서 하나 이상의 기본 테이블로부터 유도된 가상 테이블

07-기-1 ❖VIEW 특징

07-산-1

07-산-3

07-기-1

07-산-2

- 구조가 기본테이블과 거의 유사 하므로 **삽입,삭제,갱신의 명령도 table을 따름**
- 물리적으로 구현되지 않았다.
- 논리적 독립성 제공
- 필요한 데이터로만 구성 -> 관리 수월, 명령 간단
- 데이터 보호 -> 효율적
- 삽입, 삭제, 갱신 연산이 가능
- 다른 VIEW 정의에 기초
- 하나의 VIEW를 삭제 -> 그 VIEW를 기초로 만들어진 VIEW도 자동 삭제

❖VIEW의 장.단점

- 장점
 - 논리적 데이터 독립성
 - 사용자 데이터 관리 용이
 - 자동 보안
- 단점
 - 독립적인 인덱스를 가질 수 없다.
 - VIEW의 정의 변경(Alter VIEW) 불가
 - 삽입, 삭제, 갱신 제한

❖트랜잭션 정의

-데이터베이스의 상태를 변화시키는 논리적 연산의 집합

07-기-1 ❖트랜잭션 특징

07-기-2

07-기-3

- 원자성(Atomicity) : 모두 반영되거나 아니면 전혀 반영되지 않아야 된다.
- 일관성(Consistency): 트랜잭션이 그 실행을 성공적으로 완료하면 언제나 일관성 있게 DB 상태로 변환
- 독립성,격리성(isolation): 둘 이상의 트랜잭션이 동시에 병행 실행되고 있을 때 또 다른 하나의 트랜잭션의 연산이 끼어들 수 없다.
- 영속성(Durability): 트랜잭션의 결과는 영구적으로 반영 cf>무결성 X

07-기-1 ❖병행제어 기법

07-기-2

07-기-3

- 로킹 (Locking) : 하나의 트랜잭션이 데이터를 액세스하는 동안 다른 트랜잭션이 그 데이터 항목을 액세스할 수 없도록 하는 방법
- 로킹 단위 : 병행제어에서 한꺼번에 로킹 할 수 있는 단위
 - 로킹 단위가 크면...
 - 병행제어 기법이 간단하다.
 - 로크의 수가 적어진다.
 - 극단적인 경우 순차처리 하는 것과 같다.
 - 로킹 단위가 작으면 -> 로크수가 커진다. -> 관리가 복잡해지고 병행 ↑
- 2단계 로킹(twophase locking)
 - 직렬성보장, 확장단계와 축소단계의 두가지(phase)가 있다, 각 트랜잭션의 로크요청과 해제요청을 2단계로 실시한다.

07-기-2 ❖분산 데이터베이스의 특징

- 07-기-3 -자료 공유 용이
- 시스템 성능 향상
- 점증적 시스템 용량 확장 용이
- 구축 비용 증가, 소프트웨어 개발 비용이 많다.
- 지역자치성이 높다
- 효용성과 융통성이 높다

07-기-1 ❖자료를 기억장치 내에 저장하는 방법

- 07-기-2 -선형 구조 : 순차 리스트 (스택, 큐, 데크, 배열), 연결 리스트
- 07-산-1 -비선형 구조 : Tree, Graph

07-산-3

❖선형구조(Linear Structure)

- 순차 리스트 (선형, Sequential List) : 연속적인 기억장소에 저장
 - 아파트를 계단으로 연속적으로 이동
 - 특징 : 구조 간단, 기억장소 이용 효율이 높음, 삽입/삭제 어려움, 검색 빠름.
- 07-기-3 연결 리스트 (Linked List) : 비연속적으로 저장
 - 아파트를 엘리베이터로 비연속적으로 이동 (포인터)
 - 특징 : 기억장소 이용 효율이 낮음, 삽입/삭제 용이, 검색 느림.

07-기-1 ❖Stack (선형 구조 > 순차 리스트의 종류)

- 07-산-1 -삽입/삭제가 한 쪽에서 이루어지는 데이터 구조 (LIFO : Last In First Out)
- 07-산-3 -Top Point : 가장 최근에 삽입된 자료 또는 가장 먼저 삭제될 자료를 가리키는 스택 포인터로 삽입일때는 Top 값 증가, 삭제일때는 Top 값 감소

❖Queue

- 노드의 삽입 작업은 선형 리스트의 한 쪽 끝에서, 제거 작업은 다른 쪽 끝에서 수행되는 자료 구조 (FIFO : First In First Out)

07-기-3 운영체제의 작업 스케줄링, 키보드 버퍼 이용 시, 스펴(spool) 운용 시

07-산-2 ❖Deque

- 07-기-3 -삽입과 삭제가 리스트의 양쪽 끝에서 발생할 수 있는 자료 구조

07-기-1 ❖삽입 알고리즘

```

Top = Top + 1
If (Top > M) Then
    Stack_overflow
Else
    Stack(top) ← data
    
```

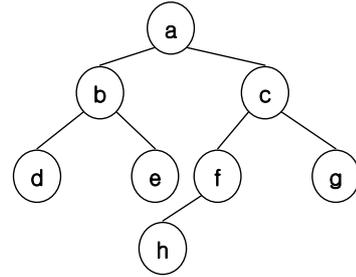
❖삭제 알고리즘

```

If (Top = 0) Then
    Stack_Empty
Else
    data ← Stack(top)
    top = top - 1
    
```

❖이진 Tree 운영법

- preorder (전위) : Root -> Left -> Right
- inorder (중위) : Left -> Root -> Right
- postorder (후위) : Left -> Right -> Root



07-기-2 ❖Tree 운영의 예

- preorder (전위) : a, b, d, e, c, f, h, g
- inorder (중위) : d, b, e, a, h, f, c, g
- postorder (후위) : d, e, b, h, f, g, c, a

❖수식표기법(prefix, infix, postfix)계산문제

07-기-1 중위 표기법(infix)의 수식 $(A+B)*C+(D+E)$ 을 후위 표기법(postfix)은?

07-기-3 다음의 Infix로 표현된 수식을 Postfix 표기로 옳게 변환한 것은?

$A = (B - C) * D + E$

07-산-2 다음의 중위(infix) 표기식을 전위(prefix) 표기식으로 옳게 변환한 것은?

$A * B + C - D / E$

07-산-2 다음 Infix 표기법을 Postfix 표기법으로 옳게 변환한 것은?

$A*(B-C)*D$

07-산-1 후위 표기(postfix)식이 다음과 같을 때 식의 계산 값은?

문1> $4\ 2\ 3\ 4\ \times\ +\ -$

문2> $5\ 3\ 4\ 5\ *\ +\ -$

❖선택정렬(Selection Sort)

- 점수 100 70 90 80 90
-> 70 100 90 80 90 (Pass 1)
 -> 70 80 100 90 90 (Pass 2)
 -> 70 80 90 100 90 (Pass 3)
 -> 70 80 90 90 100 (Pass 4)

07-기-2 ❖버블정렬(Bubble Sort)

- 점수 100 70 90 80 90
-> 70 90 80 90 100 (Pass 1)
 -> 70 80 90 90 100 (Pass 2)
 -> 70 80 90 90 100 (Pass 3)
 -> 70 80 90 90 100 (Pass 4)

07-산-1 ❖삽입 정렬(insertion Sort)

- 07-산-3 점수 100 70 90 80 90
-> 70: 100 90 80 90 (Pass 1)
 -> 70 90: 100 80 90 (Pass 2)
 -> 70 80 90: 100 90 (Pass 3)
 -> 70 80 90 90: 100 (Pass 4)

07-산-1 ❖논리회로의 분류

- 조합 논리회로 : (기억능력 X, 입력신호에 의해서만 출력 결정, gate 집합)
반가산기, 전가산기, 디코더, 인코더, 멀티플렉서, 디멀티플렉서
- 순서 논리회로 : (기억능력 O, 입력신호와 현재신호에 의해 출력 결정, gate + Flip Flop 집합)
카운터

❖Flip Flop (f/f) : 1 bit 를 기억할 수 있는 기억장치

❖관련문제

- 07-산-3 플립플롭(Flip-Flop) 회로의 설명으로 틀린 것은? [다]
- 가. 1비트의 정보량을 기억하는 기능을 가진다.
 - 나. 레지스터의 구성 회로로 널리 사용된다.
 - 다. 대표적인 조합 논리회로에 속한다.
 - 라. 어느 한 상태에서 다른 상태로 동작하기 위해서는 외부의 영향이 작용하여야 한다.
- 07-산-2 다음 중 플립플롭으로 구성할 수 없는 것은? [라]
- 가. counter 나. Register
 - 다. RAM 라. 주파수 판별기

07-기-2 ❖반가산기(Half Adder)

- 07-산-2
- 2진수 1 자리의 덧셈기
 - 하나의 AND회로와 EX-OR 회로를 조합한 회로

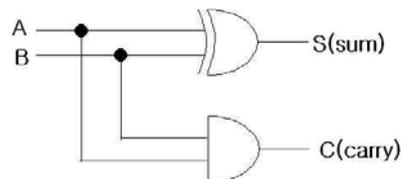
❖진리표

| A | B | S | C |
|---|---|---|---|
| 0 | 0 | 0 | 0 |
| 0 | 1 | 1 | 0 |
| 1 | 0 | 1 | 0 |
| 1 | 1 | 0 | 1 |

❖논리식

- $S = A' \cdot B + A \cdot B' = A \oplus B$
- $C = A \cdot B$

❖논리회로



07-산-3 ❖ 디코더(Decoder, 해독기)

07-기-3

-암호형태로 전달된 정보를 원래대로 복원 (人 (암호) → 기계어(원신호))

07-산-3

-n개의 입력선, 2ⁿ개의 출력선

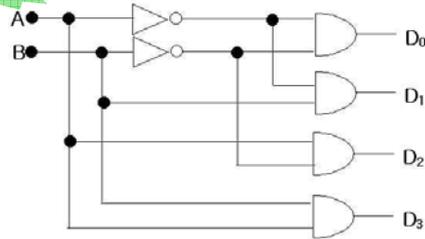
-AND gate로 구성

07-산-3 ❖ (2 X 4 디코더)

| A | B | D0 | D1 | D2 | D3 |
|---|---|----|----|----|----|
| 0 | 0 | 1 | 0 | 0 | 0 |
| 0 | 1 | 0 | 1 | 0 | 0 |
| 1 | 0 | 0 | 0 | 1 | 0 |
| 1 | 1 | 0 | 0 | 0 | 1 |

07-산-2

07-기-3



❖ 자료의 단위

-Bit < Byte < Word < Field < Record < File < Database

07-산-1

❖ Bit

07-산-2

-Binary Digit 의 약자, 정보표현의 최소단위, 0 또는 1을 나타냄

❖ Nibble

-4bit

❖ Byte

-8bit(256가지) / 문자 표현 최소단위 / 주소 지정 단위

07-기-2

❖ Word

-컴퓨터가 한번에 처리할 수 있는 명령의 단위

-Half-Word = 2byte

-Full-Word = 4byte

-Double-Word = 8byte

❖ 관련문제

07-산-3

-컴퓨터 주기억장치의 용량이 256MB라면 주소 버스의 폭은 최소한 몇 bit 이어야 하는가? [다]

가. 24

나. 26

다. 28

라. 30

07-산-3

-한글 2바이트 조합형 코드에서 한글과 영문을 구분하기 위한 비트수는? [가]

가. 1비트

나. 2비트

다. 3비트

라. 4비트

07-산-1 ❖패리티 검사 : 오류 검출(O), 정정(X)

07-산-2 ❖해밍 code : 오류 검출, 정정 가능

❖8421 code (BCD code)

10진수 1자리 -> 2진수 4자리(bit)

07-산-1 ❖3초과 code(Excess-3)

8421코드 + (3)10진수 / 비가중치, 자기보수 코드

07-기-3 ❖그레이 (Gray) code

07-산-1 BCD 코드의 인접한 자리를 XOR 연산으로 만든 코드
입출력 장치, D/A 변환기, 제어계통

❖관련문제

07-산-3 742'1'코드 표현에 의한 십진수 6의 값은? [다]
가. 0110 나. 1100 다.1001 라.1011

07-산-3 10진수 634를 BCD code로 표현 하였을 때 옳은것은? [가]
-가. 0110 0011 0100 나. 0110 0011 0011
-다. 0011 0011 0100 라. 0011 0011 0011

07-기-3 ❖제어장치 (CU : Control Unit)

07-기-2 모든 장치 지시, 제어 (제어기능)

07-기-1 주기억장치에 기억된 명령을 꺼내서 해독하고, 시스템 전체에 지시 신호를 내는 것은?

❖산술논리연산장치(ALU : Arithmetic Logic Unit)

실제 연산하는 장치 (연산기능)

❖레지스터

CPU 속에서 일시적으로 값을 기억하는 임시기억장소 (기억기능)

종류

PC (Program Counter) : 다음에 실행할 명령의 번지 기억

IR (Instruction Register) : 현재 수행 중인 명령의 내용 기억

07-산-1 07-기-3 ACC (Accumulator 누산기) : 연산의 결과를 일시적으로 저장

07-기-2 MAR (Memory Address Register) : 데이터의 번지를 저장

MBR (Memory Buffer Register) : 기억장치에서 참조한 데이터를 저장

PSR (Program Status Register) : CPU 상태 레지스터 -> 상태 정보 (PSW)

07-산-2 ❖버스

장치들 간 상호 필요한 정보를 교환하기 위해 연결하는 공동의 전송선 (전달기능)

입출력 장치와 주기억 장치를 연결하는 중개 역할을 담당하는 부분

- Address Bus, Data Bus, Control Bus

❖ 관련문제

07-기-3 = 16-bit 컴퓨터 시스템에서 다음과 같은 2가지의 명령어 형식을 사용할 때 최대 연산자 수는? [나]
 가. 64 나. 72
 다. 86 라. 144



07-기-3 = 명령어 수행시간이 10ns 이고, 명령어 패치시간이 5ns, 명령어 준비시간이 3ns 이라면 인스트럭션의 성능은 얼마인가? [라]
 가. 0.1 나. 0.3 다. 0.5 라. 1.25

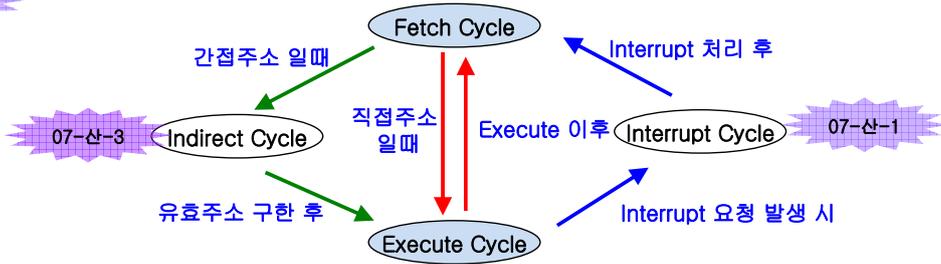
07-산-2 = 다음 3가지의 연산자(operator)가 혼합되어 나오는 식에서 시행(연산) 순서는? (단, 가장 왼쪽에 기술된 것이 가장 우선순위가 높다.) [다]

- ① 관계 연산자(Relative operator) ② 논리 연산자(Logical operator)
- ③ 산술 연산자(Arithmetic operator)

가. ①→②→③ 나. ②→①→③
 다. ③→①→② 라. ①→③→②

07-산-2 = 주기억장치의 용량이 512KB인 컴퓨터에서 32비트의 가상주소를 사용하는데 페이지의 크기가 1K워드이고 1워드가 4바이트라면 주기억장치의 페이지 수는 몇 개인가? [다]
 가. 32개 나. 64개 다. 128개 라. 512개

07-기-1 ❖ 메이저 스테이트의 4가지 상태도
07-산-1



❖ Fetch

- 명령어를 주기억장치에서 중앙처리장치의 명령레지스터(IR)로 가져와 해독하는 단계

07-산-2 ❖ Indirect

- Fetch 단계에서 해석된 명령의 주소부(operand)가 간접주소인 경우 수행되며 오퍼랜드의 주소를 읽는다.

07-기-2 ❖ Execute

- Fetch 단계에서 인출하여 해석한 명령(연산)을 실행하는 단계

❖ Interrupt

- 인터럽트 발생시 복귀주소(PC)를 저장시키고, 제어순서를 인터럽트 처리 프로그램의 첫 번째 명령으로 옮기는 단계

❖ 관련문제

07-산-1 21. CPU가 직접 제어하는 방식 중에서 입.출력 장치의 요구가 있을 때 데이터를 전송하는 제어 방식은? [나]

가. 프로그램 입.출력 제어 방식

나. 인터럽트 입.출력 제어 방식

다. 채널에 의한 입.출력 제어 방식

07-기-3 07-산-1 ❖ Interrupt의 정의

- Computer system에 예기치 않은 일이 발생하였을 때 제어프로그램에게 알려주는 것

07-기-1 ❖ 인터럽트 수행 순서

- Interrupt 요청신호 발생
- 현재 수행 중인 명령을 완료하고, 상태 기억 (복귀주소 : M/M 0번지, Stack M)
- Interrupt 판별
- ISR 에 의해 Interrupt 처리
- 보존한 프로그램 상태 복구 후 계속 처리

07-기-1 ❖ Interrupt 종류

- 내부(Internal) 인터럽트 : 프로그램에 의한 인터럽트 (0 으로 나눔, 무한 루프)
- 07-기-2 07-기-3 - 외부(External) 인터럽트
 - 정전 (Power Fail) : 우선 순위가 가장 높음
 - Timer 에 의한 인터럽트, - 입출력 인터럽트
- 소프트웨어 인터럽트
 - SVC(SuperVisor Call) 인터럽트: 입출력 수행, 기억장치 할당 및 오퍼레이터와의 대화

07-기-1 07-기-2 ❖ DMA(Direct Memory Access)

- DMA는 Direct Memory Access의 약자이다.
- DMA는 기억장치와 주변장치 사이의 직접적인 데이터 전송을 제공한다.
- DMA는 블록으로 대용량의 데이터를 전송할 수 있다.
- 기억 소자와 I/O 장치 간의 정보 교환 때 CPU의 개입 없이 직접 정보 교환이 이루어질 수 있는 방식

- 07-산-1 **우선순위 판별 방법**
- 07-산-3 -S/W : 폴링(Polling) - 요청신호 플래그
- 07-기-1 -H/W : 데이지 체인, 직렬 - 장치번호 버스
- 인터럽트를 발생하는 모든 장치들을 인터럽트의 우선순위에 따라 직렬로 연결함으로써 이루어지는 우선순위 인터럽트 처리방법

❖ 관련문제

- 07-기-3 39. 우선순위 인터럽트 운영 방식이 아닌 것은? [라]
 - 가. LCFS(Last Come First Service) 나. FCFS(First Come First Service)
 - 다. Masking Scheme 라. Fixed Service
- 07-기-3 40. 소프트웨어에 의한 우선 순위 체제의 특성을 설명한 것으로 옳지 않은 것은? [라]
 - 가. 경제적이다. 나. 융통성이 있다.
 - 다. 반응속도가 느리다. 라. 정보량이 매우 적은 시스템에 적합하다.
- 07-산-3 24. 인터럽트 반응 시간(Interrupt response time)에 대한 설명으로 옳은 것은? [가]
 - 가. 인터럽트 요청 신호를 발생한 후부터 인터럽트 취급 루틴의 수행이 시작될 때까지이다.
 - 나. 일반적으로 하드웨어에 의한 방식이 소프트웨어에 의한 처리보다 느리다.
 - 다. 인터럽트 반응 속도는 하드웨어나 소프트웨어에 필요한 기억 공간에 의한 영향이 없다.
 - 라. 인터럽트 요청 신호의 발생 후부터 취급 루틴의 수행이 완료될 때까지의 시간이다.

- 07-산-1 **ROM(Read Only Memory)**
- Only Read, 비 휘발성, 입.출력 시스템의 자가 진단 프로그램 저장
- 종류
 - Mask ROM : 공장에서 내용이 기입된다.
 - 07-산-3 PROM : PROM Writer로 기입 (내용을 지울 수 없다.)
 - EPRAM : 자외선을 이용 (내용 지울 수 있다.)

❖ RAM(Random Access Memory)

- 현재 사용 중인 프로그램, 데이터 저장, 휘발성
- 07-기-1 종류 : 재충전의 유무에 따라서

| 구 분 | DRAM (동적) | SRAM (정적) |
|------|------------|-------------------|
| 구성소자 | 콘덴서 | 플립플롭 |
| 특 성 | 주기적 재충전 | 전원공급 되는 동안만 기억 유지 |
| 전력소모 | 적다 | 많다 |
| 접근속도 | 느리다 | 빠르다 |
| 가 격 | 저가 | 고가 |
| 용 도 | 일반적인 주기억장치 | 캐시 메모리 |

❖ 관련문제

- 07-산-3 입력 주소 선이 10개, 출력 데이터 선이 8개인 ROM의 기억 용량은? [나]
 - 가. 256 byte 나. 1024 byte 다. 2048 byte 라. 8192 byte
- 07-기-1 DRAM의 사이클 타임(Mt)과 기억장치 접근시간(At)의 관계식으로 옳은 것은? [가]
 - 가. Mt >= At 나. Mt = At 다. Mt <= At 라. Mt < At

07-산-1 07-기-1

❖ 자기 테이프

- 순차접근만 가능한 기억장치
- 자기 테이프 Record 크기가 80자로서 블록(Block)의 크기가 2,400자일 경우 블록 팩터(Block Factor) 또는 블록화인수는 30이 된다.

07-산-3

❖ 자기 디스크

- 직접+순차 접근 모두 가능한 방식
- 자기디스크의 접근시간
 - Access Time : word 단위 정보를 읽거나 쓰는데 걸리는 시간
 - Access Time = Seek Time + Search Time + Transmission Time
- 07-산-3 Seek Time(탐색 시간) : 트랙을 찾는데 걸리는 시간, 액세스 암(Access arm)이 이동하는 시간
- Search Time(회전 지연 시간) : 섹터를 찾는데 걸리는 시간
- Transmission Time(전송 시간) : 해당 내용 전송

1. 인터프리터(interpreter)를 사용하는 언어는? [가]
 - 가. BASIC 나. FORTRAN
 - 다. PASCAL 라. Machine Code
2. 중앙처리장치의 기억 모듈에 중복적인 데이터 접근을 방지하기 위해서 연속된 데이터 또는 명령어들을 기억 장치 모듈에 순차적으로 번갈아 가면서 처리하는 방식은? [나]
 - 가. 복수 모듈 나. 인터리빙
 - 다. 멀티플렉서 라. 셀렉터
3. 다음 중 2의 보수(2's complement) 가산 회로로서 정수 곱셈을 이행할 경우 필요 없는 것은? [라]
 - 가. shift 나. add
 - 다. complement 라. Nomalize
4. 고급 언어(High-Level Language)에 대한 특징으로 가장 옳은 것은? [나]
 - 가. Computer 하드웨어와 Compiler에 종속적이다.
 - 나. Computer 하드웨어에 독립적이고, Compiler에 종속적이다.
 - 다. Computer 하드웨어에 종속적이고, Compiler에 독립적이다.
 - 라. Computer 하드웨어와 Compiler에 독립적이다.

❖ 운영체제(OS; Operation System)의 개념

- 컴퓨터 시스템을 구성하고 있는 하드웨어 장치와, 일반 컴퓨터 사용자 또는 컴퓨터에서 실행되는 응용 프로그램의 중간에 위치하여 사용자들이 보다 쉽고 간편하게 컴퓨터 시스템을 이용할 수 있도록 제어 관리하는 프로그램

07-산-1
07-기-3

❖ 운영체제의 성능 평가 요인

- 응답 시간(Turn Around Time)
 - 컴퓨터에 명령을 내린 후 결과를 얻을 때까지 걸리는 시간으로 수치가 낮을수록 좋음
- 처리량(Throughput)
 - 단위 시간 내에 처리할 수 있는 일의 양으로 수치가 높을수록 좋음
- 신뢰도(Reliability)
 - 시스템이 고장 없이 주어진 기능을 정확하게 수행하는 것으로 수치가 높을수록 좋음
- 사용 가능도(Availability)
 - 각 사용자의 시스템 자원 요구시, 제공할 수 있는 실제 시스템자원의 사용 가능 시간으로 수치가 높을수록 좋음

07-기-3

❖ 제어프로그램(Control Unit)

- 감시 프로그램(Supervisor Program), 자료 관리 프로그램(Data Management Program), 작업 관리 프로그램(Job Management Program)

07-산-2

❖ 처리프로그램(Process Program)

- 언어번역 프로그램(Language Translator Program), 서비스 프로그램(Service Program), 문제 처리 프로그램(Problem Processing Program)

07-기-1

❖ 운영체제 목적

07-기-2
07-산-1
07-산-2
07-산-3

- 사용자와 컴퓨터간의 인터페이스 제공
- 자원의 효율적인 운영 (프로세서, 기억장치, 주변장치, 파일 관리)
- 데이터 공유 및 주변장치 관리
- 처리 능력, 신뢰도, 사용 가능도 향상
- 응답시간, 반환시간 단축

07-기-3

❖ 운영체제의 발달 과정

07-산-3

- 일괄 처리 시스템(Batch Processing System)
 - 자료를 1주일 또는 1개월 등의 기간 단위로 모아 두었다가 처리하는 방식
- 다중 프로그래밍 시스템(Multi Programming System)
 - 동시에 2개 이상의 프로그램을 주기억 장치에 기억시켜 놓고 하나의 프로세서가 고속으로 처리하는 방식
- 시분할 처리 시스템(Time Sharing System)
 - 한 시스템을 여러 명의 사용자가 공유하여 동시에 작업을 수행하는 방식
- 다중 처리 시스템(Multi Programming System)
 - 하나의 컴퓨터에 2개 이상의 CPU가 메모리 장치와 입출력 장치를 공유하여 프로그램을 처리하는 방식
- 실시간 처리 시스템(Real Time Processing System)
 - 데이터가 발생할 때마다 컴퓨터로 처리하여 은행의 On-line 예금처럼 즉시 그 결과를 나타내도록 하는 방식
- 다중 모드 처리
- 분산 처리 시스템(Distributed Processing System)

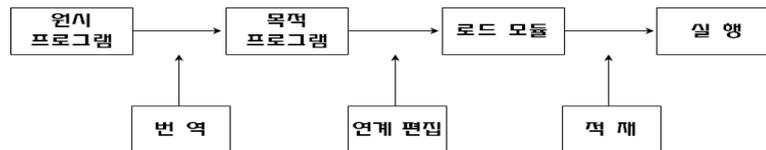
❖ **저급언어 : 기계중심의 언어**

- 기계어 : 2진수만 사용하여 명령어와 데이터를 표현한 기계 중심 언어
- 어셈블리어 : 기계 중심의 언어로 기계어에 1:1 대응되는 언어

❖ **고급언어 : 인간중심의 언어**

- FORTRAN : 과학기술용
- COBOL : 사무처리용
- LISP : 인공지능 분야에 사용되는 언어
- SNOBOL4 : 스트림 연산
- BASIC : 대화형(회화형)언어
- C 언어 : 시스템 프로그래밍 언어, 이식성이 높은 언어, 구조적 프로그래밍이 가능

❖ **언어의 번역과정**



❖ **로더의 기능**

- 할당(Allocation), 연결(Link), 재배치(Relocation), 적재(Load)
- compile-and-go 로더 : 번역기가 로더의 역할까지 담당 (번역+로더)
- 절대(Absolute) 로더 : 적재 기능만 하는 간단한 로더 - (할당, 연결-프로그래머, 재배치-언어번역기)

07-기-1 ❖ **프로세스(Process) 정의**

- PCB를 가진 프로그램
- 실 기억장치에 저장된 프로그램 (실행중인 프로그램) 07-기-2
- 프로세서가 할당되는 실체
- 프로시저(프로그램 일부)가 활동 중인 것
- 지정된 결과를 얻기 위한 일련의 동작
- 운영체제가 관리하는 실행단위
- 비동기적(비연속적 처리)인 활동 주체
- 보조기억장치에 있는 파일은 프로세스(X) 07-산-1

07-기-1 ❖ **매크로 프로세서**

- 처리 과정 : 매크로 정의 인식 → 매크로 정의 저장 → 매크로 호출 인식 → 매크로 호출 확장
- 특징 : 매크로 내에 매크로를 정의할 수 있다.
- 매크로(개방형)와 부프로그램(폐쇄형) 차이점 : 매크로 내용 삽입 → M/M 절약 X, 실행 빠르다.

07-산-2 ❖ **PCB (Process Control Block)**

- 운영체제가 프로세스에 대한 중요한 정보를 저장해 놓은 곳 (프로세스 정보 리스트)
- 저장 정보 : 프로세스 현재상태, 포인터, 프로세스 고유 식별자, 스케줄링 및 프로세스 우선순위, CPU 레지스터 정보, 주기억장치 관리정보, 입출력상태 정보, 계정정보

07-기-2 ❖ 스케줄링의 정의

- 컴퓨터 시스템의 성능을 높이기 위해 그 사용 순서를 결정하기 위한 정책

07-기-2 ❖ 스케줄링의 목적(성능 평가)

07-산-3

- 처리율 증가, CPU 이용률 증가, 우선 순위 제도, 오버헤드(부하) 최소화, 응답 시간 / 반환 시간 / 대기 시간 최소화, 균형 있는 자원의 사용, 무한 연기 회피
- 프로세서 스케줄링 : 프로세스가 실행되지 위해 CPU를 할당 받는 시기와 특정 프로세스를 지정하는 작업 (단기 스케줄링)

❖ 스케줄링의 기법

07-산-3

- 비선점 스케줄링 (Non Preemptive) : 비효율적
 - 프로세스에게 이미 할당된 CPU를 강제로 빼앗을 수 없고, 사용이 끝날 때까지 기다려야 하는 방법
 - 일괄 처리(오버헤드 발생 X), 실시간 처리가 안되므로 중요한 작업이 기다리는 경우 발생
- 선점 스케줄링 (양보) : 효율적
 - 우선 순위가 높은 다른 프로세스가 할당된 CPU를 강제로 빼앗을 수 있는 방법
 - 실시간 처리, 대화식 시분할 처리(오버헤드 발생 O)

❖ FIFO (First-In First-Out) = FCFS

- 대기 큐(ready queue)에서 도착한 순서에 따라 디스패치 됨
- 평균 반환시간 증가

07-기-3 ❖ SJF (Shortest Job First)

- 작업이 끝나기까지의 실행 시간 추정치가 가장 작은 작업을 먼저 실행

07-기-1 ❖ HRN (Highest response ratio Next)

07-산-2

07-산-3

- 대기시간 또는 서비스 시간에 따른 우선 순위를 결정하는 기법
- 우선순위 계산식 : (대기 시간+서비스 시간)/서비스 시간
- SJF 방법의 단점을 보완하여 개발한 것으로, 프로그램의 처리 순서는 그 실행(서비스) 시간의 길이뿐만 아니라 대기시간에 따라 결정되는 스케줄링 방식

❖ 우선순위

- 대기 큐에서 기다리는 각 프로세스마다 우선 순위를 부여하여 그 중 가장 높은 프로세스에게 먼저 CPU를 할당하는 기법

❖ 기한부

- 프로세스에게 일정한 시간을 주어 그 시간 안에 프로세스를 완료하도록 하는 기법

❖교착상태(Dead Lock) : 예측 못한 다운

- 상호 배제에 의해 나타나는 문제점으로, 둘 이상의 프로세스들이 자원을 점유한 상태에서 서로 다른 프로세스가 점유하고 있는 자원을 요구하며 무한정 기다리는 현상 (무한연기 X -> Aging 기법)

❖교착상태 발생 4가지 필요충분 조건 (동시 만족 시 발생)

07-기-2
07-기-3
07-산-1

- 상호배제(Mutual Exclusion) : 한 번에 한 개의 프로세스만이 공유 자원을 사용할 수 있어야 함
- 점유와 대기(Hold & Wait) : 최소한 하나의 자원을 점유하고 있으면서 다른 프로세스에 할당되어 사용되고 있는 자원을 추가로 점유하기 위해 대기하는 프로세스가 있어야 함
- 비선점(Non-preemptive) : 다른 프로세스에 할당된 자원은 사용이 끝날 때까지 강제로 빼앗을 수 없어야 함 (비양보 상태)
- 환형대기(Circular Wait) : 공유 자원과 공유 자원을 사용하기 위해 대기하는 프로세스들이 원형으로 구성되어 있어 자신에게 할당된 자원을 점유하면서 앞이나 뒤에 있는 프로세스의 자원을 요구해야 함

❖교착상태 해결방안

- 예방 기법(Prevention)
- 회피 기법(Avoidance) : 은행원 알고리즘(Banker's Algorithm)이 사용
- 발견 기법(Detection)
- 회복 기법(Recovery)

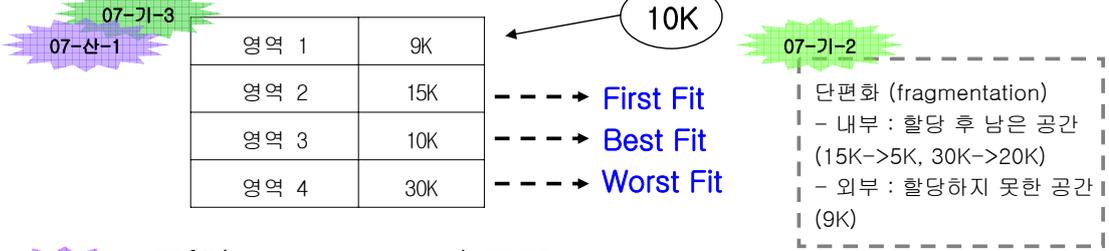
07-산-2

❖반입(Fetch) 전략

- 보조기억장치의 프로그램이나 데이터를 언제 주기억장치로 적재할 것인지를 결정

07-산-2 ❖배치(Placement) 전략

- 주기억장치의 어디에 위치시킬 것인지를 결정
- 최초 적합(First Fit) : 첫 번째 배치시키는 방법 (속도↑, 공간↓)
- 최적 적합(Best Fit) : 단편화를 가장 작게 남기는 분할 영역에 배치시키는 방법 (속도↓, 공간↑)
- 최악 적합(Worst Fit) : 단편화를 가장 많이 남기는 분할 영역에 배치시키는 방법 (속도↓, 공간↓)



07-산-3 ❖교체(Replacement) 전략

- 주기억장치의 모든 영역이 이미 사용중인 상태에서 주기억장치에 배치하려고 할 때, 이미 사용되고 있는 영역 중에서 어느 영역을 교체하여 사용할 것인지를 결정 (FIFO,OPT,LRU,LFU,NUR,SCR)

❖종류

- OPT (OPTimal replacement, 최적교체), FIFO (First In First Out), LRU (Least Recently Used), LFU (Least Frequently Used), NUR (Not Used Recently)

07-기-3 ❖FIFO (First In First Out)

- 가장 먼저 들어온 페이지를 먼저 교체시키는 방법
- 벨레이디의 모순(Belady's Anomaly) 현상 : 페이지 프레임 수가 증가하면 페이지 부재가 더 증가

❖OPT (OPTimal replacement) 최적교체

- 앞으로 가장 오랫동안 사용하지 않을 페이지를 교체하는 기법 (실현 가능성X)

07-기-1 ❖LRU (Least Recently Used) 최근 사용이 가장 적은

- 최근에 가장 오랫동안 사용하지 않은 페이지를 교체하는 기법

07-산-2 ❖LFU (Least Frequently Used) 사용 횟수가 가장 적은

- 사용 빈도가 가장 적은 페이지를 교체하는 기법

07-산-1 ❖NUR (Not Used Recently) 최근 사용하지 않은

- 최근에 사용하지 않은 페이지를 교체하는 기법
- 각 페이지마다 2개의 하드웨어 비트(호출 비트, 변형 비트)가 사용됨

07-기-2 ❖페이지 크기가 작을 경우 (10K -> 1K)

- 페이지 수 증가 -> 페이지 맵핑 테이블 커진다 -> 맵핑 속도 느리고 기억 공간 낭비 발생
- 디스크 접근 횟수 증가 -> 전체적인 입.출력 시간은 늘어남

07-기-1 ❖ 워킹 셋 (Working Set)

- 프로세스가 일정 시간 동안 자주 참조하는 페이지들의 집합으로, 자주 참조되는 워킹 셋을
- 주기억장치에 상주시킴으로써 페이지 부재 및 페이지 교체 현상을 줄임

07-기-3 ❖ 스래싱 (Thrashing)

- 프로세스의 처리 시간보다 페이지 교체 시간이 더 많아지는 현상 -> CPU이용률 저하
- 워킹 셋은 스래싱 방지 방법 중 하나

07-산-1 ❖ 구역성 (Locality, 국부성) - 캐시 메모리 이론

- 프로세스가 실행되는 동안 일부 페이지만 집중적으로 참조하는 성질

❖ 시간 구역성

- 최근에 참조된 기억 장소가 가까운 장래에도 계속 참조될 가능성이 높음
- 예) Loop(반복), 스택, 부프로그램(Sub Routine), Counting(1씩 증감), Totaling(집계)

07-기-1 ❖ 공간 구역성

- 하나의 기억 장소가 참조되면 그 근처의 기억 장소가 계속 참조될 가능성이 높음
- 예) 순차적 수행, 배열 순례

07-기-3 ❖ 세그멘테이션(Segmentation) 기법

- 가상기억장치에 보관되어 있는 프로그램을 다양한 크기의 논리적인 단위로 나눈 후 주기억장치에 적재시켜 실행시키는 기법 => 메모리 절약

❖ 정의

- 사용할 데이터가 디스크상의 여러 곳에 저장되어 있을 경우 데이터를 액세스하기 위해
- 디스크 헤드가 움직이는 경로를 결정하는 기법

❖ 목적

- 처리량의 최대화, 응답시간의 최소화, 응답시간 편차의 최소화

07-산-1 ❖ FCFS (First-Come First-Service)

- 입출력 요청 대기 큐에 들어온 순서대로 서비스를 하는 방법

07-기-1 ❖ SSTF (Shortest Seek Time First) - 가장 짧은 거리

- 탐색 거리가 가장 짧은 트랙에 대한 요청을 먼저 서비스하는 기법
- 탐색 시간 편차 (기아상태발생)

07-기-3 ❖ SCAN - 한 방향으로 가장 짧은 거리

- SSTF가 갖는 탐색 시간의 편차를 해소하기 위한 기법
- 현재 헤드의 위치에서 진행 방향이 결정되면 탐색 거리가 짧은 순서에 따라 그 방향의 모든 요청을 서비스하고, 끝까지 이동한 후 역방향의 요청 사항을 서비스함

07-기-2 ❖ C-SCAN (Circular SCAN)

- 항상 바깥쪽에서 안쪽으로 움직이면서 가장 짧은 탐색거리를 갖는 요청을 서비스

07-기-1 ❖ 파일 시스템의 기능

- 사용자와 보조기억장치 사이에서 인터페이스를 제공
- 사용자가 파일을 생성, 수정, 제거 할 수 있도록 한다
- 파일을 안전하게 사용, 보호하는 기능

07-기-2 ❖ 파일 디스크립터 (File descriptor) = FCB (파일 제어 블록)

07-산-1

- 파일을 관리하기 위한 시스템이 필요로 하는 파일에 대한 정보를 갖는 제어 블록 => 사용자 직접 참조 X

07-산-2

- 보통 보조기억장치에 저장되었다가 파일이 오픈 될 때 주기억장치로 전달
- 정보 : 파일이름, 위치, 구조, 액세스 제어정보, 파일유형, 날짜, 시간, 액세스 횟수 등

❖ 디렉토리 구조

- 1단계 : 모든 파일이 하나의 디렉토리 내에 위치
- 2단계 : 마스터 / 사용자 파일 디렉토리
- 트리구조 : 루트 / 종속(서브) 디렉토리, DOS, Windows, UNIX 에서 사용

❖ 비순환 그래프 구조

- 공용(O), 사이클(X), 디스크 공간을 절약할 수 있음
- 하나의 파일이나 디렉토리가 여러 개의 경로 이름을 가질 수 있음

07-기-3 ❖ 일반 그래프 구조

- 트리 구조에 링크(Link)를 추가 -> 순환(O)
- 사이클이 허용되고, 불필요한 파일제거를 위해 참조
- 카운터가 필요한 디렉토리 구조

07-기-3 ❖ 자원 보호 기법

- 접근 제어 행렬 : 자원 보호의 일반적인 모델, 객체에 대한 접근 권한을 행렬로써 표시
- 전역 테이블
- 접근 제어 리스트
- 권한 리스트

07-산-1

❖ 파일 보호 기법

- 파일의 명명(Naming) : 파일 이름을 모르는 사용자를 접근 대상에서 제외시키는 기법
- 비밀번호>Password, 암호) : 각 파일에 판독 암호와 기록 암호를 부여하여 암호를 아는 사용자에게만 접근을 허용하는 기법
- 접근 제어(Access Control) : 사용자에 따라 공유 데이터에 접근할 수 있는 권한을 제한하는 방법

07-기-1

07-기-2

07-산-3 ❖ 보안

- 외부 보안
 - 시설보안 : 천재지변과 외부 침입자로부터 보안
 - 운용보안 : 관리, 정책 등의 여러 통제 절차를 통해 보안
- 내부 보안 : 하드웨어나 운영체제의 내장된 기능
- 사용자 인터페이스 보안
 - 사용자의 신원을 운영체제가 확인하고 나서 접근하도록 하는 방법
 - 패스워드

07-기-2 ❖ Master/Slave(주/종) 처리기

07-기-3

07-산-3

- 주 프로세서 : 입출력과 연산 담당, 운영체제를 수행
- 종 프로세서 : 연산만 담당, 사용자 프로그램만 담당
- 주 프로세서가 고장 나면 전체 시스템 다운

❖ 분리 수행 처리기

- 주/종 처리기의 비대칭성을 보완하여 각 프로세서가 독자적인 운영체제를 가짐

❖ 대칭적 처리기

- 분리 실행 처리기 구조의 문제점을 보완한 것으로, 여러 프로세서들이 완전한 기능을 갖는 하나의 운영체제를 공유

07-기-1 ❖ 분산운영체제의 특징

07-기-2

07-기-2

07-산-1

07-산-2

- 과부하를 줄일 수 있고, 점진적 확장(특정한 시스템 병목 현상을 제거하기 위해 필요한 자원을 추가할 수 있으므로 선택적인 성능 향상을 가능하게 한다.) 가능
- 투명성
- 소프트웨어 개발이 어렵다
- 보안문제가 발생한다

❖ 완전 연결 (Fully Connection)형 = 망형

- 각 사이트들이 시스템 내의 다른 모든 사이트들과 직접 연결된 구조
- 기본 비용은 많이 들지만 통신 비용은 적게 들고, 신뢰성이 높음

❖ 계층형 (Hierarchy) = 트리형

- 분산 처리 시스템의 가장 대표적인 형태
- 부모 사이트가 고장나면 그 자식 사이트들은 통신이 불가능함

07-산-1 ❖ 성형 = 스타형

07-산-2

07-산-3

- 모든 사이트가 하나의 중앙 사이트에 직접 연결
- 중앙 사이트가 고장 날 경우 모든 통신이 단절됨

❖ 링형 = 환형

- 인접하는 다른 두 사이트와만 직접 연결된 구조
- 정보는 단방향 또는 양방향으로 전달될 수 있음

07-기-1 ❖ 다중 접근 버스 연결(Multi Access Bus Connection)형

07-기-3

- 공유 버스에 연결된 구조
- 사이트의 고장은 다른 사이트의 통신에 영향을 주지 않지만, 버스의 고장은 전체 시스템에 영향을 줌
- 사이트의 추가와 삭제가 용이하다.

❖ UNIX 특징

- 07-기-2
- 07-기-2
- 07-산-2
- 시분할 시스템을 위해 설계된 대화식 운영체제
- 소스가 공개된 개방형 시스템 (Open System)
- 대부분 C언어로 작성 -> 이식성 / 호환성이 높음
- 크기가 작고 이해하기가 쉬우며, Multi-User, Multi-Tasking 지원
- 많은 네트워킹을 제공하므로 통신망(Network) 관리용 운영체제로 적합함
- 트리 구조의 파일 시스템으로, 전문적인 프로그램 개발에 용이함
- 다양한 유틸리티 프로그램들이 존재함

❖ UNIX 시스템의 구성

- 07-산-1
- 07-산-2
- 07-산-3
- 07-기-2
- 커널(Kernel)
 - UNIX의 가장 핵심적인 부분
 - 프로세스 관리, 기억장치 관리, 파일 관리, 입.출력 관리, 프로세스간 통신, 데이터 전송 및 변환 등 여러 가지 기능 수행
- 셸(Shell)
 - 명령어 해석기
 - 시스템과 사용자 간의 인터페이스 담당
- 유틸리티(Utility)
 - 일반 사용자가 작성한 응용 프로그램을 처리하는 데 사용

❖ UNIX 파일 시스템의 구조

- 07-기-3
- 07-기-1
- 07-기-3
- 부트 블록, 슈퍼 블록, Inode 블록, 데이터 블록
- Inode에 포함된 정보
 - 파일 소유자의 사용자 번호(UID) 및 그룹 번호(GID), 파일 크기, 파일 type, 생성 시기, 최종 수정 시기, 최근 사용 시기, 파일의 보호 권한, 파일 링크 수, 데이터가 저장된 블록의 시작, 주소 등의 정보를 가지고 있음 (파일 최초 수정 시기 X)

❖ 프로세스 관련 명령

- 07-산-1
- 07-산-3
- 07-기-3
- 07-산-3
- fork : 프로세스 생성, 복제, 호출
- exec : 프로세스 수행
- wait : 부모 프로세스를 임시 중지
- getpid : 자신의 프로세스 아이디를 얻는다
- getppid : 부모 프로세스 아이디를 얻는다
- signal : 신호를 받았을 때 프로세스가 취할 동작을 지정한다
- pipe : 프로세스 간 통신을 위한 경로를 설정
- Ps : 현재 프로세스의 상태를 확인

❖ 파일 등 관련 명령

- 07-기-1
- 07-기-1
- 07-기-1
- creat : 파일 생성
- open : 파일 준비
- cp : 파일 복사
- mv : 파일 이동, 이름 변경
- rm : 파일 삭제
- mknod : 특수 파일 생성
- mount : 새로운 파일 시스템을 서브 디렉토리에 연결
- mkfs : 파일 시스템 생성
- fsck : 파일 시스템 검사, 보수
- ls : 현재 디렉토리 내의 파일 목록 확인
- finger : 사용자 정보 표시
- chmod : 파일 사용 허가 지정 (rwx)
- cat : 파일 내용을 화면에 표시

❖정보통신의 역사

- 1세대 : 부호를 이용하여 정보를 전달하는 세대
- 2세대 : 음성을 이용하여 정보를 전달하는 시대
- 3세대 : 전자 신호 및 무선을 이용하여 정보를 전달하는 세대

❖정보통신 기술의 발전

-정보통신 시스템의 발전은 **컴퓨터 기술과 데이터 전송 기술의 결합된** 형태로 나타남

❖정보통신 기술의 분류

- 정보의 전달을 중심으로 하는 통신기술은 **단말기술, 전송기술, 교환기술로** 구분
- 컴퓨터 기술로는 정보의 가공, 처리를 담당하는 분야로, 컴퓨터 시스템의 하드웨어를 다루는 컴퓨터 제조 기술, 시스템 운용을 다루는 소프트웨어 기술과 이들을 응용하는 자동화 기술로 구분

07-산-3 ❖정보통신시스템의 특징

- 통신회선을 효율적으로 이용 가능함.
- 에러제어 방식을 사용하여 시스템의 신뢰도가 높음
- 고품질의 통신서비스를 제공함

❖정보통신 시스템의 유형

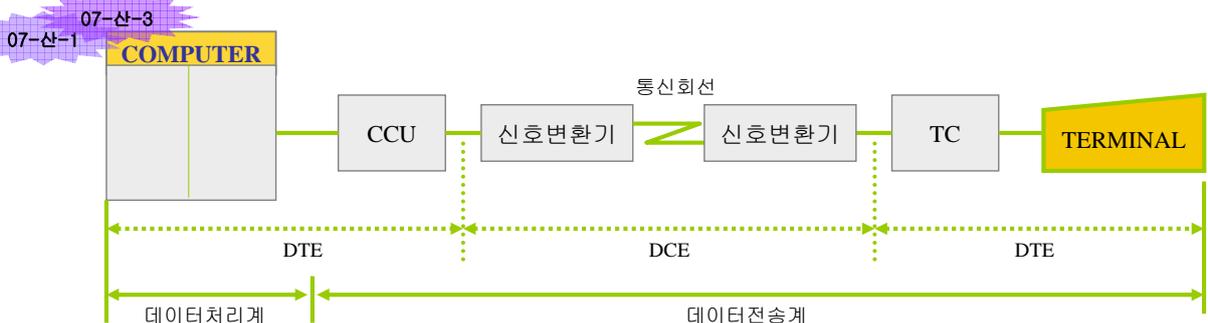
- 데이터 전송계 : 정보의 이동을 담당
- 단말 장치, 신호 변환 장치(변복조기(MODEM) 또는 DSU), 통신 회선, 통신 제어 장치 (CCU)

07-산-2 ❖데이터 처리계 : 정보의 가공, 처리, 보관 등의 기능 수행, **통신처리, 정보처리**

- 중앙 처리 장치(CPU), 주변 장치

07-산-2 ❖각부의 기능

- DTE(Data Terminal Equipment : 데이터 단말 장치)
- DCE(Data Circuit termination Equipment : 데이터 회선 종단 장치)
- CCU(Communication Control Unit : 통신 제어 장치) : 회선감시, 송수신제어, 전송오류검출
- FEP(Front End Processor : 전처리 장치)



❖ 전송기기의 정의

-DTE 간에 정보를 송.수신하기 위한 연결 전송 구간을 구성하는 요소로써 신호 변환기와 통신 회선으로 구성됨

❖ 신호변환기

-정의 : DTE간의 정보를 송수신하기 위한 연결 전송 구간을 구성하는 요소
-구성 : 신호 변환기, 통신회선

07-산-1 기능 : 변조(디지털→아날로그), 복조(아날로그→디지털):원래의 파형으로 복원

-종류

07-산-2 모뎀(Modulation DEModulation, 변복조기) : 컴퓨터의 디지털 신호를 전송회선의 아날로그로, 아날로그신호를 디지털로 바꾸어주는 변환장치

07-기-1 디지털 서비스 유닛(DSU:Digital Service Unit) : 디지털 데이터를 원래 전송에 적합한 디지털 신호로 변환하여 디지털 전송로를 통해 전송

07-산-3 -음향결합기(Acoustic Coupler) : 단말장치와 전화기를 연결하기 위한 모뎀의 일종으로 전화기의 송수화기를 음향결합기에 결합시켜 디지털 신호를 아날로그 신호로 변환하여 전송

❖ 통신회선

-단말장치와 컴퓨터를 연결하는 물리적 전송매체, 유선/무선으로 구분

❖ DTE/DCE 접속규격

-EIA에서 규정하고 있는 변복조기(MODEM)와 단말기(DTE) 사이의 연결접속방법
-ITU-T(국제 전기 통신 협회 표준 센터) : V.24

❖ RS-232C 25핀의 기능

- 2번 핀 : 송신 데이터의 신호를 취급 (TXD)
- 3번 핀 : 수신 데이터의 신호를 취급 (RXD)
- 4번 핀 : 송신요구(Request to send)
- 5번 핀 : 송신종료(Clear to send)



07-기-1 ❖ 라우팅(경로 선택) 프로토콜의 종류

07-기-2

07-기-3

- RIP (Routing Information Protocol) : 소규모 동종의 네트워크 내에서 효율적인 방법
- IGP (Interior Gateway Protocol) : 내부 게이트웨이 프로토콜
- EGP (Exterior Gateway Protocol) : 외부 게이트웨이 프로토콜
- BGP (Border Gateway Protocol) : EGP의 단점을 보완하기 위해 만든 프로토콜

❖모뎀(Modem)

-컴퓨터의 디지털 데이터를 아날로그 신호로 바꾸어 전화선(아날로그 전송로)을 통하여 송신하며, 수신할 때는 아날로그 신호를 디지털 신호로 변환

❖케이블 모뎀(Cable Modem)

-일반 모뎀에 비해 케이블 모뎀은 네트워크에 항상 연결되어 있으며, 월간 일정한 금액을 내게 되므로 많은 시간을 쓰는 사람의 경우에는 요금 면에서도 유리

❖브리지(Bridge)

-서로 독립적으로 동작하는 두 개의 근거리 통신망(LAN)을 연결, 패킷을 적절히 중계, 필터링 하는 장치

❖라우터(Router)

-동일한 전송 프로토콜을 사용하는 분리된 네트워크를 연결하는 장치, 네트워크 계층 간을 서로 연결

❖리피터(Repeater)

-근거리 통신망(LAN)의 전송 매체 상에 흐르는 신호를 정형, 증폭, 중계하는 장치

07-산-1 ❖게이트웨이(Gateway)

-다른 네트워크로 들어가는 입구 역할을 하는 네트워크 포인트

-서로 다른 프로토콜을 사용하는 망을 연결하는데 사용

07-기-2 ❖다중화의 의미

07-산-1

-여러 개의 단말기 등이 하나의 통신회선으로 데이터를 전송하는 장치

07-기-2

❖다중화기의 종류

-주파수 분할 다중화기(FDM)

-하나의 물리적 통신 채널을 여러 주파수 채널로 나누어 사용

07-기-3

-채널간의 완충 지역으로 가드 밴드(Guard band)가 필요함

-1200baud 이하의 비동기에만 사용되며, 구조가 간단함

07-기-1 시분할 다중화기(TDM)

-여러 회선들의 음성정보를 아주 작은 시간으로 나누어서 일정 순서대로 배정, PCM 방식 사용

-다중화기와 단말기의 속도 차이로 버퍼가 필요함

-주파수 분할 다중화기(FDM)보다 고속 전송이 가능

07-기-2 동기식 시분할 다중화

07-기-3

-타임슬롯을 모든 이용자에게 규칙적으로 할당

-단점 - 대역폭이 낭비

07-기-1

-비동기식 시분할 다중화

-전송할 데이터를 갖고 있는 사용자에게만 타임슬롯을 할당하고 여유 있는 슬롯은 다른 이용자에게 할당

-대역폭의 이용 효율을 높임, 통계적 시분할 다중화 방식

07-산-2 디지털 데이터의 아날로그 부호화(MODEM 이용)

-진폭편이변조(ASK : Amplitude Shift Keying) : 0과 1을 반송파의 진폭만 변화시켜 표현하는 방법

07-산-3 주파수편이변조(FSK : Frequency Shift Keying) : 0과 1을 서로 다른 주파수를 갖는 반송파로 변화시켜 표현하는 방법

07-산-3 위상편이변조(PSK : Phase Shift Keying) : 0과 1을 서로 다른 위상을 갖는 반송파로 변화시켜 표현하는 방법

-진폭위상 편이 변조(QAM or APSK)

❖아날로그 데이터의 아날로그 부호화(전화, 방송에 이용)

-진폭변조(AM : Amplitude Modulation) : 반송 신호의 진폭은 변조 신호 패턴과 함께 변함

-주파수변조(FM : Frequency Modulation) : 각각 반송 신호의 주파수 변조

-위상변조(PM : Phase Modulation) : 각각 반송 신호의 위상 변조

07-산-2 동기식과 비동기식

07-산-1

07-기-2

-동기식 : 동기문자를 이용하여 송,수신측의 동기를 유지하는 방식

-비동기식 : start bit와 stop bit를 사용하여 문자단위로 전송하는 방식, 동기식에 비해 저속이다

07-기-3

07-산-3 아날로그 데이터의 디지털 부호화(펄스 변조 방식)

07-기-3

-PCM(Pulse Code Modulation) 방식

-아날로그 정보를 디지털 정보인 펄스 부호로 변환하여 전송하고 수신측에서는 디지털 정보를 원래의 파형인 아날로그 정보로 변조시키는 방식

07-산-2 표본화 - 양자화 - 부호화의 과정 수행

07-기-3

-PAM(Pulse Amplitude Modulation) 방식

-펄스의 진폭을 변화시켜 변조시키는 방식

-PWM(Pulse Width Modulation) 방식

-펄스의 폭을 변화시켜 변조시키는 방식

-PPM(Pulse Position Modulation) 방식

-펄스의 위치를 변화시켜 변조시키는 방식

-DM, ADPCM

❖디지털데이터의 디지털 부호화(DSU에 이용)

-NRZL(NonReturn-to-Zero-Level)부호화 : PC와 외부 모뎀 사이 또는 단말기와 컴퓨터 사이와 같이 매우 짧은 연결에 사용

❖Bps

- 매 초당 전송되는 비트의 수를 나타내는 것으로, 어떤 신호가 2,400bps라 함은 1초에 2400개의 비트가 전송됨을 의미

❖Baud

- 매 초당 몇 개의 신호 변화가 있었는가를 나타내는 속도단위
- 한 비트가 한 신호단위일 경우는 Baud와 Bps는 동일하다.

07-기-1
07-기-2
07-산-1
07-산-2
07-기-3

❖Bps 와 baud와의 관계표

| 관계 단위 | 1 BIT ONEBIT 2위상 ×1(1배) | 2BIT DIBIT 4위상 ×2(2배) | 3BIT TRIBIT 8위상 ×3(3배) | 4BIT QUADBIT 16위상 ×4(4배) |
|----------|----------------------------------|--------------------------------|---------------------------------|-----------------------------------|
| BPS | | | | |
| BAUD | | | | |

❖채널 용량

- 정보가 에러 없이 그 채널을 통해 보내어질 수 있는 최대속도를 나타내며, 일반적으로 BPS(Bit Per Second)단위로 표현
- 정보의 전송 속도는 전송 매체로 정보를 실어 보내기 위해 컴퓨터 관련 기기에 의해 만들어지는 정보의 생산 속도

❖샤논(Shannon)의 법칙

-W라는 대역폭을 가진 채널이 N이라는 잡음 세력을 가졌고 이 채널에 S라는 신호 세력을 가진 신호를 전송할 때 얻을 수 있는 통신 용량 C(단위 : bps)

07-산-3 대역폭과 신호 세력이 높을수록, 잡음 세력이 낮을수록 채널 용량은 좋음

07-산-1
$$C = W \log_2 \left(1 + \frac{S}{N} \right) [bps]$$

❖ **유도(유선)매체**

-트위스티드 페어(Twisted-Pair)

-물리적 특성 : 나선 형태의 절연된 두 개의 구리선이 상호 전기적 간섭 현상을 줄이기 위해 서로 감겨있는 형태의 케이블

-전송 특성 : 다른 전송 매체에 비해 대역폭이 넓지 못하고 일반적인 특성도 뒤떨어지며, 고속 전송도 비교적 제한적

-동축 케이블(Coaxial Cable)

-물리적 특성 : 동일한 동심축을 외부의 전류로부터 보호하기 위해서 플라스틱 절연체를 씌우고 그 위에 그물 모양의 구리망을 두름

-전송 특성 : 트위스티드 페어에 비해 우수한 주파수 특성을 가지고 있으며, 혼선과 방해를 훨씬 덜 받음

07-산-3 **광섬유 케이블(Optical Fiber Cable)**

07-산-2

07-산-1

-물리적 특성 : 코어와 클래딩, 코팅 부분으로 구성되며, 빛 신호가 코어를 따라 이동하면서 클래딩에 반사되는 과정을 반복하여 데이터를 전송하는 케이블

-전송 특성 : 넓은 대역폭, 작은 크기와 무게, 적은 감쇠도, 전자기적 격리, 보다 넓은 리피터 간격과 같은 전송 특성을 가짐

-광통신의 3요소

-발광기 / 수광기 / 광섬유

07-산-3 **광손실**

-흡수손실, 접속손실, 레일레리 산란 손실

07-기-2

❖ **감쇠(Attenuation)**

-데이터를 전송 매체를 통해 원격지로 전송하고자 할 때, 전자적 신호의 세기가 거리가 멀어질수록 점차적으로 약해지는 현상

-감쇠 현상은 높은 주파수에서 더 심하며 이것이 왜곡을 초래함

❖ **왜곡(Distortion)**

-주파수 성분들의 특성에 의해 원래 신호가 다른 형태로 일그러지는 현상

-감쇠왜곡 : 아날로그의 경우 주파수에 따라 수신된 신호의 강도가 약해져 발생

-자연왜곡 : 전송 매체에서 발생하는 현상으로, 동기 맞춤 등에 의해 보상

-하모니 왜곡 : 신호의 감쇠가 신호의 진폭에 따라 달라지는 잡음

-주파수 왜곡 : 전송채널에 보내지는 원 주파수가 변형되는 현상

❖ **잡음(Noise)**

-전송로 상에서 전송신호에 유입되는 불필요한 신호

-열잡음 : 전송 매체의 저항에 의한 열의 영향 때문에 발생 (=White noise)

-누화(Cross Talk) : 인접 선로의 상호간섭에 의한 발생

-충격성 잡음(Impulse Noise) : 선로의 파괴나 손상에 의해 발생

-상호 변조 잡음 : 동일 전송 매체를 공유하는 서로 다른 주파수를 갖는 신호 사이에 발생

❖문자 위주 전송 프로토콜

- 반이중 통신 방식에 사용, P2P 또는 Multipoint에서 이용, 효율성 나쁨
- 데이터 길이 제한, Stop & Wait ARQ 방식 사용

07-기-1 **BSC 프로토콜**

- 데이터 링크 제어 문자
 - ENQ(ENquiry) : 송신을 받을 준비가 되었는지 확인(상대방의 응답을 요구)
 - ACK(ACKnowledge) : 수신한 정보 메시지에 대한 긍정 응답 신호 (↔NAK)

07-기-1 **STX(Start of TeXt) : 전송할 메시지의 시작임을 알림**

- DLE(Data Link Escape) : 제어 문자 앞에 놓여 이들 문자들이 유효한 제어 문자임을 표시

❖비트 위주 전송 프로토콜

- 특수한 플래그 문자가 메시지의 처음과 끝에 위치함
- 제어 비트를 각 비트 워칭서 유도
- ISO에서 현재 세계적으로 널리 사용되는 표준
- HDLC(High-level Data Link Control) 프로토콜이 대표적**
- Go back N ARQ 방식 사용

❖바이트 위주 전송 프로토콜

- 헤더의 처음을 표시하는 특수문자와 메시지부를 구성하는 문자 개수, 메시지의 수신 상태를 표시하는 제어 정보와 블록 체크를 포함시켜 전송하는 방식
- DDCMP 프로토콜**

❖HDLC 국의 종류

- 주국, 보조국, 복합국

❖전송프레임

07-기-1 **플래그 - 주소필드 - 제어필드 - 정보필드 - 프레임 검사 순서 - 플래그**

07-기-2

07-산-1

07-산-3

- 플래그(Flag) : 시작과 종료를 표시(01111110),
- 주소부(Address) : 프레임을 수신하거나 송신하는 부스테이션을 식별하는데 사용
- 제어부(Control) : 정보, 감시, 비번호제 프레임으로 구성
- 정보부(Information) : 사용자 사이에서 교환되는 정보 메시지와 제어정보가 들어있는 부분
- 프레임 검사 부분(FCS : Frame Check Sequence) : 주소부, 제어부, 정보부의 내용이 오류없이 상대방에 정확히 전송되었는가를 확인

07-기-3 07-산-2 **❖HDLC 프로토콜의 특징**

- 비트지향형의 프로토콜, 제어부의 확장이 가능, 데이터링크 계층의 프로토콜

07-산-2 **❖HDLC 데이터 전달모드**

- 정규응답모드, 비동기 균형모드, 비동기 응답모드

❖유사 프로토콜

- SDLC(Synchronous Data Link Control) : HDLC의 변형된 형식 사용, 명령과 응답은 루프 토플로지를 제공, 루프 또는 원형 폴링 동작 수행 제공

❖패리티 검사(Parity Check)

-한 블록의 데이터 끝에 패리티 비트(Parity Bit)를 추가하는 것으로, 가장 간단한 에러 검출 기법

❖블록 합 검사(BSC; Block Sum Check)

-각 문자당 패리티 체크 비트와 전송 프레임의 모든 문자들에 대한 패리티 문자를 함께 전송하는 방식
-패리티 검사의 단점 보완

07-산-3 ❖순환 잉여도 검사(CRC; Cyclic Redundancy Check)

-에러 검출코드인 FCS(Frame Check Sequence)에 사용
-집단 에러에 신뢰성 있는 다항식 코드를 사용하는 에러 검출 방식
-여러 개의 에러 검출과 수정이 가능

❖자동 재전송 방식(ARQ : Automatic Repeat reQuest)

-에러 검출 후에 송신측에게 에러가 발생한 데이터 블록을 다시 전송해 주도록 요청함으로써 에러 정정

❖ARQ의 종류

-정지 대기(Stop and Wait) ARQ : 송신측은 한 블록을 전송한 다음 수신측에서 에러 발생 검출에 의해 역채널을 통해서 ACK(정상수신)나 NAK(에러발생) 신호를 보내올때까지 기다리는 방식

07-산-1 연속적 ARQ : 한블록씩이 아니라 연속적으로 블록을 전송하는 방식

07-기-3 -Go back N 방식 : 에러 발생이후 모든 블록 재전송

-선택적(Selective) ARQ : 에러발생 블록만 재전송하는 방법으로 수신측에서 수신된 블록들을 모아 재조립해야 하므로 더 복잡함

-적응적(Adaptive) ARQ : 채널 효율의 최대화를 위해 블록 길이를 동적으로 변경

❖전진 에러 수정(FEC : Forward Error Correction) 방식

07-산-1 에러 검출과 수정을 동시에 수행하는 방식(해밍코드, 상승 부호 코드, 길쌈부호(CONVOLUTION))

07-산-3 -길쌈부호 : 한 신호를 기준으로 다른 신호가 상대적인 시간 변화를 가지며 그에 따른 적분 계산을 통해 산출된 신호(서로 다른 두개의 신호를 합성하는 과정)

-ARQ에 비해 역채널이 불필요하고 연속적인 데이터 흐름이 가능

07-산-3 **프로토콜(Protocol)의 정의**

-통신시스템에서 멀리 있는 다른 통신 시스템과 전송 매체를 통하여 통신을 할 수 있도록 해주는 절차나 규범

07-산-1 **프로토콜의 기본적 구성 요소**

07-산-2

- 구문(Syntax)** : 데이터의 형식이나, 부호화 및 신호레벨 규정
- 의미(Semantics)** : 해당 패턴 해석과 전송의 조정, 오류 제어
- 순서(Timing)** : 접속 개체간의 통신 속도 조정이나 메시지 순서 규정

07-산-3 **계층 프로토콜의 목적**

- 복잡한 네트워크를 논리적으로 좀더 작고 이해하기 쉬운 부분(계층)으로 나눔
- 네트워크 기능 간의 표준화된 인터페이스 제공(소프트웨어 프로그램 모듈 간의 표준 인터페이스)
- 표준화된 인터페이스를 이용하여 각 계층 간 독립적인 구현 가능
- 모듈화에 의한 전체 설계가 용이하다.
- 이기종간 호환성 유지가 비교적 쉽다.
- 한 계층을 수정할 때 다른 계층에 영향을 주지 않는다.

07-기-1 **데이터 링크 제어가 통신 채널을 관리하는 단계**

07-기-2

-회선연결 - 링크확립 - 정보전송 - 링크절단 - 회선절단

07-기-3

07-기-3 **폴링 시스템(Polling System)**

-폴 명령이 주국에서 보조국으로 보내지며, 폴이 의미하는 것은 실제로 보조국이 주국으로 보낼 데이터를 갖고 있는지를 물어 보는 것

셀렉션 시스템(Selection System)

-주국이 보조국에게 데이터를 전송하려고 할 경우, 데이터 수신 준비 여부를 확인하는 방식

프로토콜의 방식

- BSC 프로토콜 : 문자(Character) 방식의 프로토콜(Loop 방식이 불가능함)
- DDCMP 프로토콜 : 바이트(Byte) 방식의 프로토콜
- SDLC 프로토콜 : 비트(Bit) 방식의 프로토콜
- HDLC 프로토콜 : 비트(Bit) 방식의 프로토콜, 데이터 링크 계층에서 가장 효율이 좋은 프로토콜 방식, 단방향(Simplex), 반이중(Half Duplex), 전이중(Full Duplex) 통신 방식 모두를 지원. 포인트 투 포인트, 멀티 포인트, 루프 방식을 지원

| | |
|------------------|---|
| 문자(Character) 방식 | 전송 데이터의 처음과 끝에 동기를 위한 전송 제어 문자를 포함시켜 전송하는 방식(BSC 프로토콜이 속함) |
| 바이트(Byte) 방식 | 전송 데이터의 헤더에 데이터의 문자 수, 메시지 수신 상태 등의 제어 정보를 포함시켜 전송하는 방식(DDCMP 프로토콜이 속함) |
| 비트(Bit) 방식 | 전송 데이터의 처음과 끝에 특수한 플래그 문자를 포함시켜 전송하는 방식(SDLC, HDLC 프로토콜이 속함) |

07-기-3 ❖OSI(개방형 시스템간의 상호접속) 7계층

- 상위 계층 : 응용 계층(최상위 계층), 표현 계층, 세션 계층, 전송 계층
- 하위 계층 : 네트워크 계층, 데이터 링크 계층, 물리 계층(최하위 계층)

| | | | |
|----------------------------|---|------------------|---|
| | 7 | 응용 계층 | 사용자가 OSI 환경을 이용할 수 있는 서비스 제공 |
| 07-산-1 | 6 | 표현 계층 | 암호화, 내용 압축, 형식 변환 등의 기능을 제공 |
| | 5 | 세션 계층 | 응용들 사이의 연결을 확립.유지.단절시키는 수단을 제공 |
| 07-기-2 07-기-2 07-기-1 | 4 | 트랜스포트 계층 | 링크 종점 간에 정확한 데이터 전송을 제공, TCP, UDP |
| 07-산-2 07-산-3 | 3 | 네트워크 계층 | 경로 제어와 흐름 제어를 수행, IP프로토콜 |
| 07-기-1 07-기-2 | 2 | 데이터 링크 계층 | 동기화, 에러제어, 흐름제어 기능을 담당 |
| 07-산-1 | 1 | 물리 계층 | 실제 회선 연결을 확립.유지.단절하기 위한 기계적, 전기적, 기능적, 절차적 특성 정의 |

07-산-1 07-산-2 흐름제어(flow control) : 패킷수를 적절히 조절하여 전체시스템의 안정성을 기하는 기능,오버플로우 예방

❖ISO(International Organization for Standardization)

-국제 표준화 기구

07-산-2 ❖ITU-T

-구 CCITT(Consultative Committee for Integration Telegraph and Telephone)가 개편된 국제전기통신연합인 ITU의 산하기관으로 1956년에 창설

-V 시리즈 : 아날로그(전화망) 통신에서 사용되는 인터페이스를 위한 권고안

07-기-1 -X 시리즈 : 데이터 통신에서 사용되는 인터페이스를 위한 권고안

❖ANSI : 미국 표준안 제정 기관

-민간인에 의한 임의의 국가 규격 제정 기관

-제조업자나 공중통신 사업자 등으로 구성

❖EIA(Electronics Industries Association) : 미국 전자 공업 협회

-데이터 통신 규격을 다루는 기술 위원회

-RS-232 인터페이스 규격을 제정

❖IEC : 전기 전자 분야의 표준화를 주도

❖IEEE(미국 전기 전자 공학회) : LAN 표준을 규정하고 있는 조직

07-산-2 -DQDB [distributed queue dual bus] : 미국전기전자학회(IEEE)에서 제정한 도시권 통신망(MAN)의 표준 규격인 IEEE 802.6에 채용되어 있는 다중 접속 프로토콜.

07-기-1 ❖ 회선 교환 방식

07-기-3

- 데이터를 전송하기 전에 컴퓨터와 단말기 사이에 통신 회선(전송 회선)을 설정한 통신 정보를 송수신하는 방식을 취하는 교환망
- 회선 교환(Circuit Switching) 방식은 컴퓨터와 터미널 간에 통신 회선을 고정적으로 할당하여 데이터를 교환하는 방식

07-산-1 ❖ 패킷 교환 방식

07-산-3

07-기-2

07-기-3

- 전송하고자 하는 데이터를 보관해 두었다가 패킷 단위로 분해한 후 전송하는 방식
- 패킷 교환(Packet Switching) 방식은 패킷 형태로 만들어진 일정 길이의 전송 단위 데이터를 송신측 패킷 교환기에 기억시켰다가 수신측 주소에 따라 적당한 통신 경로를 선택하여 수신측 패킷 교환기에 전송하는 방식

07-기-1 장애발생시 대체경로 선택이 가능하다.

07-기-2

- 가상회선방식(연결지향형)과 다이어그램방식으로 세분화 됨
- 회선교환방식과의 공통점은 별도의 호(Call)설정 과정이 있다는 것이다

07-산-1

07-기-2

❖ 메시지 교환 방식

- 축적 후 전달 또는 전문 교환형 통신망으로, 기억 장치에 정보를 기억해 두었다가 단말기에게 전송하는 방식
- 메시지 교환(Message Switching)방식은 교환기가 송신측의 메시지를 받아서 수신측에 보내는 방식으로, 전송하는 메시지 길이의 제약없이 전송하는 방식

❖ LAN의 개요

- 근거리 통신망(LAN ; Local Area Network)은 제한된 일정 지역 내에 설치된 통신망으로, 약 10km 이내의 거리에서 100Mbps 이내의 고속 데이터 전송이 수행되는 시스템

07-산-2 ❖ LAN의 특징

07-산-1

- 단일 건물 내에 설치되고, 패킷 지연이 최소화
- 경로 설정이 불필요하고, 확장성과 재배치가 용이
- 광대역 전송 매체의 사용으로 인하여 고속 통신이 가능
- OSI 참조모델의 하위층에서 사용됨

❖ LAN의 효과

- 정보 자원의 공유, 정보의 실시간 처리, 정보 자원의 일관성
- 통합된 사무자동화 구축, 이기종 간의 통신, 정보처리 시스템의 비용 절감

07-기-1 ❖ LAN에서 사용되는 매체 액세스 제어(Access Control)기법

- TOKEN-BUS, CSMA/CD(이더넷에서 채택), TOKEN-RING

07-기-3

07-산-1 ❖ VAN(Value Added Network)의 개념 및 정의

07-산-3

07-기-1

- 통신 회선을 기간 통신 사업자로부터 임차하여 시설망을 구축하고 이를 이용, 축적해 놓은 정보를 유통시키는 부가가치 통신망
- 공중 통신 회선에 교환설비, 컴퓨터 및 단말기 등을 접속시켜 새로운 부가 기능을 제공하는 통신망

07-기-3 ❖성형(Star)

- 중앙에 컴퓨터나 교환기가 있고, 그 주위에 단말장치들을 분산시켜 연결시킨 형태
- 노드의 추가 및 증설이 용이하며 고장발견이 쉬움
- 잠재적 병목성, 중앙 지역고장의 경우 전 시스템 사용 불가

❖망형(Mesh)

- 모든 단말기와 단말기들을 통신회선으로 연결시킨 형태
- 통신회선 장애시 다른 경로를 통하여 데이터 전송 가능
- 통신회선의 총 길이가 가장 김, 통신회선의 수 = $n(n-1)/2$ (n : node의 수)

❖링형(Ring)

- 서로 이웃하는 단말기들끼리만 연결한 방식
- 비용이 적게 들고 병목 현상이 적음, 통신망의 재구성이나 변경이 어려움

07-기-2 ❖버스형(Bus)

- 1개의 통신 회선에 여러대의 단말 장치를 접속하는 형식
- 관리가 용이하고 새로운 노드의 삽입이 용이
- 통신 채널이 한 개여서 고장시 네트워크 전체가 동작하지 않으므로 잉여채널필요

❖트리형(Tree)

- 중앙의 최상위 컴퓨터를 중심으로 일정한 지역의 단말기까지 연결하고 다시 연장되는 형태
- 제어가 간단, 관리와 확장이 용이
- 분산처리 시스템 가능, 중앙지점에서 병목 현상 발생 가능

❖WAN(Wide Area Network)의 정의

- WAN은 물리적으로 각기 다른 지역에 설치되어 있는 DTE 혹은 LAN을 연결해 주는 능력을 갖는 네트워크를 의미

❖ISDN(Integrated Services Digital Network)

- 음성을 비롯하여 화상 및 데이터 서비스를 통합하여 제공할 수 있는 종합 정보 통신망으로, 원 신호의 신호 대역폭에 따라 협대역 ISDN(N-ISDN)과 광대역 ISDN(B-ISDN)으로 구분할 수 있음

07-산-3

- B-ISDN(Broadband-ISDN, 광대역 ISDN)** : 영상의회의, 화면 전송 등과 같이 연속성의 실시간 또는 군집 데이터 신호를 넓은 대역에 걸쳐 광대역으로 실현하기 위해 만들, ATM(Asynchronous Transfer Mode)방식, 광전송 등 초고속 전송기술이 필요, 1.544[Mbps]이상의 고속데이터 및 영상서비스

❖ISDN 채널의 사용 용도

- B채널 : 64Kbps이하의 사용자 정보를 전송하며 회선교환, 패킷교환, 디지털 전용회선방식에 의해 사용자 정보를 전송
- D채널 : 16Kbps이하의 패킷 교환 방식에 의한 사용자 정보 전송에 사용
- H채널 : B채널을 통해 제공하는 모든 방식의 정보 전송을 보다 고속으로 제공

❖ISDN 서비스

- 베어러 서비스(Bearer Service) : OSI의 하위 계층에 해당되는 정보 전달 서비스
- 텔레 서비스(Tele-Service) : 단말기와 통신망의 기능으로 제공되는 총체적 서비스로 ISDN망을 이용하여 단말기 등이 제공하는 서비스
- 부가 서비스(Supplementary Service) : 음성, 데이터, 영상 등의 기본 서비스에 추가되어 새로운 서비스 기능

❖ **셀룰러(Cellular) 시스템**

- 서비스 지역의 제한과 가입자 수용 용량의 한계를 극복하기 위하여 제안된 개념
- 공간적으로 분포하는 채널수를 증가시켜 충분한 가입자를 확보할 수 있도록 하는 이동통신 방식
- 주파수 재사용, 셀 분할 기법

07-산-2

- 핸드오프**(통화 중인 가입자가 새로운 기지국 서비스 지역으로 진입할 때도 통화의 단절 없이 계속 통화가 가능)
- 로밍(이동전화 가입자가 타 교환국에 있어도 이동전화 서비스를 받을 수 있음)의 특징이 있음

❖ **셀룰러 시스템의 구성**

- PSTN(일반적인 전화교환국에 설치된 교환기)
- MTSO(이동전화 교환국)
- BS(기지국)
- MS(모바일 스테이션)
- PS(이동국)

❖ **PCS 시스템**

- PCS(Personal Communication Services)는 일반 가입자가 휴대 단말기를 이용해 언제, 어디서나, 누구와도 통신할 수 있도록 하는 고도의 개인 휴대통신 서비스로서, 기존의 이동전화보다 한 단계 진화된 형태의 이동통신 서비스

❖ **IMT-2000**

- 전 세계적 표준화 및 동일 주파수를 사용하여 하나의 단말기로, 전 세계 어디서나 누구 하고나 어떠한 형태의 통신도 가능함

❖ **인터넷의 개요**

- 전 세계의 컴퓨터들이 연결되어 있는 컴퓨터망으로 시간적, 공간적으로 제약 없는 통신이 가능한 지구촌 통신망. **TCP/IP를 기본 프로토콜로 사용하여** 접속된 네트워크의 집합체. **미 국방성의 ARPANET**는 인터넷의 모체가 되는 네트워크

❖ **인터넷의 주소**

- TCP/IP : 인터넷에서 LAN으로 구축된 통신망 간의 상호 통신 규약
- IP 주소 : 인터넷에 연결된 컴퓨터가 갖는 고유한 주소. 현재 사용하는 IP주소 체계는 **IPv4로 32비트 체계, 차세대 IP주소 체계는 IPv6, 32비트에서 128비트로 확장**

07-기-2

❖ **TCP(Transmission Control Protocol) / IP(Internet Protocol)의 개요**

- TCP : 인터넷에서의 데이터 전송을 제어하는 프로토콜
- IP : 패킷을 목적지까지 전송하는 프로토콜, **네트워크 계층에 해당**
- TCP/IP는 TCP와 IP같은 특정 프로토콜만을 의미하는 것이 아니라 OSI의 네트워크 계층, 전송계층, 응용 계층에서 사용되는 프로토콜 조합을 의미
- 두 종단간에 연결을 설정한 후 데이터를 전송하는 접속형(Connection-Oriented) 프로토콜

07-기-3

❖ **TCP헤더에 포함되는 정보 : 긴급포인터, 순서번호, 체크섬**

❖ **UDP의 개요**

- 종단 간에 연결을 설정하지 않은 채 데이터를 전송하는 비접속형 전송 서비스를 제공하는 프로토콜
- TCP는 UDP에 비해 뛰어난 신뢰성을 갖지만, 양단 간에 접속을 유지해야 하고, 초기 접속시 TCP간 접속을 설정하는 시간 소비로 인해 **멀티미디어 전송에는 주로 UDP가 사용됨**

❖ IP주소 체계

| IP 주소의 종류 | IP 주소의 처음 값 | 특징 |
|-----------|-------------|------------|
| A 클래스 | 1~127 | 국가, 대형 통신망 |
| B 클래스 | 128~191 | 중대형 통신망 |
| C 클래스 | 192~223 | 소규모 회사 |

❖ 도메인 이름

- IP 주소는 숫자로 되어 있는 것을 문자로 이용하여 사용자가 알기 쉽게 표기하는 주소

07-기-3 ❖ DNS(Domain Name System)

- DNS의 개요
- 인터넷에 연결되는 모든 컴퓨터들은 고유한 주소인 IP 주소 할당
- 일반 사용자들은 기억하기 쉬운 이름을 IP에 할당해서 사용
- 도메인 이름을 IP주소로 상호 변환하는 기능

| 발행처 | 도메인 및 의미 |
|-----------|--|
| NIC(미국) | · com(영리단체) / org(비영리단체) · net(네트워크 관리기관) |
| KRNIC(한국) | co.kr(영리단체) / or.kr(비영리단체) / ne.kr(네트워크 관리기관) pe.kr(개인) / ac.kr(전문대 이상) / hs.kr(고등학교) ms.kr(중학교) / es.kr(초등학교) / sc.kr(특수학교) |

07-산-2

❖ 핑(Ping)

- 사용자가 접속하려고 하는 호스트가 운영되고 있는지를 확인할 수 있는 서비스

❖ 핑거(Finger)

- 특정 인터넷 사용자의 사용한 시간, 접속 여부 등에 관한 정보를 파악할 수 있는 서비스

07-기-2 ❖ 원격 접속(Telnet : 텔넷)

- 하나의 컴퓨터를 네트워크를 통하여 다른 컴퓨터의 단말기로 만든 소프트웨어를 먼 거리에 있는 컴퓨터를 자신의 컴퓨터처럼 사용할 수 있는 서비스

❖ 파일 전송 프로토콜(FTP : File Transfer Protocol)

- 인터넷에 연결된 컴퓨터에 존재하는 파일을 송수신 할 수 있는 서비스

❖ 전자 우편(E-mail)

- E-mail을 송신하기 위한 SMTP(Simple Mail Transfer Protocol) 프로토콜과 E-mail를 수신하기 위한 POP3(Post Office Protocol3) 프로토콜이 지원

❖ 소프트웨어공학(Software Engineering)

-가장 경제적으로 신뢰도 높은 S/W를 만들기 위한 방법, 도구와 절차들의 체계화 한 학문

❖ S/W (프로젝트) 개발 절차

-요구 분석 -> 설계 -> 구현(코딩) -> 테스트(시험) -> 유지보수

07-기-2 ❖ 소프트웨어 위기

- 소프트웨어 개발 속도가 하드웨어 개발 속도를 따라가지 못해 소프트웨어에 대한 사용자들의 요구사항을 처리할 수 없는 문제가 발생함을 의미
- 소프트웨어 공학이 나타나게 된 배경이 됨

07-기-3 ❖ 좋은 소프트웨어의 조건

07-기-2

- 남이 알아보기 쉬워야 한다.
- 경제적, 문서화가 잘 되어 있어야 한다.
- 사용법, 구조의 설명, 성능, 기능이 이해하기 쉬워야 한다.
- 유지보수가 용이해야 한다.
- 사용자 수준에 따른 적당한 사용자 인터페이스를 제공해야 한다.

❖ 소프트웨어 생명 주기 단계

- 정의단계 : 타당성 검토 단계, 계획 단계, 요구사항분석 단계
- 개발단계 : 설계 단계, 구현 단계, 테스트 단계

07-기-2 유지보수단계 : 가장 비용이 많이 요구되는 단계

07-기-3

07-기-3 ❖ 폭포수 모형 : 순차적(고정) -> 요구분석 불만족

- 가장 오래되고 폭넓게 사용된 전통적인 소프트웨어 생명주기
- 물이 위에서 아래로 떨어지듯이 단계가 순차적으로 진행되고 단계별 정의가 분명
- 두 개 이상의 과정이 병행 수행되거나 이전 단계로 넘어가는 경우가 없음
- 개발 과정 중에 발생하는 새로운 요구나 경험을 설계에 반영하기 힘들(요구사항 변경 X)
- 제품의 일부가 될 매뉴얼 작성 필요
- 각 단계가 끝난 후 결과물이 명확히 나옴

07-기-1 ❖ 프로토타입 모형 : 모형(가변) -> 요구분석 만족

07-기-3

- 실제 상황이 나오기 전에 가상으로 시뮬레이션을 통해 최종 결과물에 대한 예측을 할 수 있음
- 개발 단계에서 오류 수정을 할 수 있음
- 요구사항을 충실히 반영할 수 있지만 비용 증가됨

❖ 나선형 모형 : 폭포수 장점 + 프로토타입 장점

- 점증적 생명주기 모델
- 위험분석 단계에 초점
- Boehm(보험) 제안
- 비용이 많이 들고, 시간이 많이 소요되지만 완성도 높으므로 대규모 프로젝트에 유리

❖ 4GT (4th Generation Techniques) : 4세대 기법

- 4세대 언어(비주얼 베이직) 이용 -> 원시 코드를 자동으로 생성
- 설계 단계 단축 -> 개발 시간 감소 (소규모 개발 시 효율적)

❖ **프로젝트 관리 : 계획대로 완료될 수 있도록 관리**

- 주어진 기간 내에 최소의 비용으로 사용자를 만족시키는 시스템 개발
- 소프트웨어 생명 주기 전 과정에 걸쳐 진행됨

07-기-1
07-기-3

❖ **효과적인 프로젝트 관리를 위한 3대 요소 (3P)**

- 사람(people) : 인적 자원
- 문제(problem) : 문제 인식
- 프로세스(process) : 작업 계획

❖ **중앙 집중형 (책임 프로그래머 팀)**

- 한 사람에 의하여 통제 -> 소규모 프로젝트 적합
- 책임 프로그래머 : 분석, 설계, 작업 지시 등 모든 기술적 판단
- 보조 프로그래머 : 책임 프로그래머 업무 지원
- 프로그래머 : 코딩, 검사, 디버깅, 문서 작성 등
- 프로그램 사서 : 프로그램 리스트, 설계 문서, 검사 계획 등

07-기-2

❖ **분산형 (민주주의적) : 링 모양 구조**

- 모든 팀 구성원이 동등한 위치에서 의사 결정 -> 장기 프로젝트 적합
- 서로의 일을 검토하고 결과에 대해 같은 그룹의 일원으로 책임짐
- 구성원의 작업 만족도가 높고, 이직률이 낮음

❖ **프로젝트 계획 수립 : 수행 전 예측하는 작업**

- 프로젝트가 수행되기 전에 소프트웨어 개발 영역(범위) 결정, 필요한 자원, 비용, 일정 등을 예측

07-기-3

❖ **프로젝트 계획 수립 시 소프트웨어 영역 결정 사항**

- 기능, 성능, 제한조건, 신뢰도, 위험성(X)

07-기-2
07-기-3

❖ **브룩스(Brooks) 법칙**

- 새로운 개발 인력이 진행 중인 프로젝트에 투입될 경우 작업 적응 기간과 부작용으로 인해 빠른시간 내에 프로젝트는 완료될 수 없다.

07-기-3

❖ **정형 기술 검토 (FTR : Formal Technical Review)**

- 가장 일반적인 검토 방법으로 소프트웨어 품질 보증 활동
- 목적 : 기능과 로직의 오류 발견, 사용자 요구사항의 확인, 프로젝트 관리의 편리성 등
- 지침 사항 : 의제 제한성, 논쟁과 반박의 제한성, 제품 검토의 집중성, 참가 인원의 제한성 등

07-기-2

❖ **검토 회의(워크스루, Walkthrough)**

- 개발 각 단계에서 개최하는 기술 평가 회의
- 오류 검출에 초점을 두고 해결책은 나중에 미루고, 검토 회의 후에 해결
- 발견된 오류는 문서화
- 검토를 위한 자료를 사전에 배포하여 검토

07-기-2 **❖ 품질 관리**

-어떤 항목이나 제품이 설정된 기술적 요구사항과 일치하는가를 적절하게 확인하는데 필요한 체계적이고도 계획적인 유형의 활동

❖ 품질 목표의 항목

- 정확성 : 사용자 요구 기능 충족 정도
- 07-기-2 -신뢰성 : 옳고 일관된 결과를 얻기 위해 요구되는 기능
- 이식성 : 다른 H/W 환경에서 운용 가능
- 상호 운용성 : 다른 S/W와 정보를 교환할 수 있는 기능
- 유지보수성 : 변경 시 수정에 대한 노력의 최소화 정도

- 07-기-1 -효율성 : 기능 수행 시 필요한 자원의 소요 정도
- 무결성 : 허용되지 않는 사용이나 자료의 변경을 제어

- 07-기-3 -사용 용이성 : 사용하기 쉬운 정도
- 유연성 : 쉽게 수정할 수 있는 정도
- 시험 용이성 : 평가를 쉽게 해 주는 정도
- 재사용성 : 전체나 일부 소프트웨어가 다른 응용 목적으로 사용 가능
- 모듈화, 상품화, 최적화 (X)

07-기-1 07-기-2 **❖ 위험 관리**

-프로젝트 추진 과정에서 예상되는 각종 돌발 상황을 미리 예상하고 이에 대한 적절한 대책을 수립하는 일련의 활동

07-기-1 **❖ 가장 대표적인 위험 요소 : 사용자 요구 사항 변경**

❖ 소프트웨어 개발방법론

-정의 : 과거 경험을 토대로 성공적으로 평가되는 소프트웨어를 분석 및 설계방법들을 모아 하나의 개발방법으로 정형화 한 것

- 07-기-1 -요구사항 분석 기법 : 사용자 면접, 현재 사용 중인 문서 검토, 설문 조사를 통한 의견 수렴

❖ 구조적 분석 기법(도구)

-자료의 흐름과 처리를 중심으로 하는 요구사항 분석 방법
-종류 : 자료 흐름도, 자료 사전, 소단위 명세서, 개체 관계도, 상태 전이도

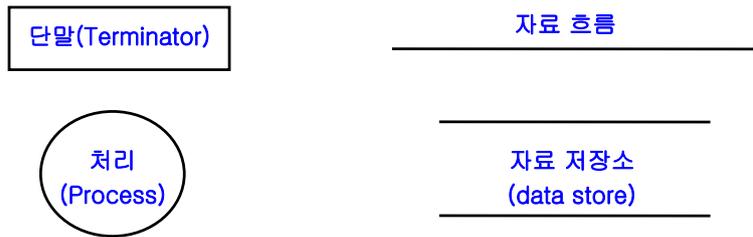
07-기-2 **❖ 소프트웨어 형상 관리**

-형상 정의 : 소프트웨어 개발 단계의 각 과정에서 만들어지는 프로그램, 문서, 데이터 등을 통칭

❖ 형상 관리

-개발 과정의 변화되는 사항을 관리하는 것
-소프트웨어 개발과정에서 소프트웨어의 변경사항을 관리하기 위해 개발된 활동
-소프트웨어 개발의 전 단계에 적용되는 활동, 유지보수 단계에서 수행

07-기-1 ❖ 자료흐름도 (DFD : Data Flow Diagram)
 07-기-2
 07-기-3



- 구조적 분석 도구
- 시스템 내의 모든 자료 흐름은 4가지의 기본 기호로 표시
- 각각의 변환(처리)에 대하여 개별적인 상세화가 가능

❖ 자료사전 (DD : Data Dictionary)

- DFD에 있는 자료를 더 자세히 정의하고 기록한 것
- 데이터를 설명하는 데이터 (메타 데이터)

| 기 호 | 의 미 | 기 호 | 의 미 |
|-----|-------------------|-------|------------------|
| = | 정의 | [] | 선택 "Optional" |
| + | 연결 | () | 생략 |
| { } | 반복 "Iteration Of" | * * | 주석, 설명 "Comment" |

07-기-3 07-기-2

07-기-1 ❖ 요구 분석용 CASE (자동화 도구)
 07-기-2

- 요구사항을 자동으로 분석하고, 요구사항 분석 명세서를 기술하는 도구
- 소프트웨어 생명 주기의 전체 단계를 연결해 주고 자동 해주는 통합된 두고를 제공한다.
- 개발 과정의 속도를 향상 시킨다.
- 소프트웨어 부품의 재사용을 가능하게 한다.
- 종류 : SREM, PSL/PSA, EPOS

07-기-1 ❖ 결합도 (Coupling): 모듈 간에 상호 의존도

- 독립적인 모듈이 되기 위해서는 **결합도가 약해야 함**
- 종류 : 데이터 < 스탬프 < 제어 < 외부 < 공통 < 내용

07-기-2 ❖ 응집도 (Cohesion) : 모듈 안의 요소들이 서로 관련된 정도

- 모듈이 독립적인 기능으로 잘 정의되어 있는 정도
- 독립적인 모듈이 되기 위해서는 **응집도가 강해야 함**
- 종류 : 우연적 < 논리적 < 시간적 < 절차적 < 교환적 < 순차적 < 기능적

07-기-1 ❖ N-S 차트 (Nassi-Schneiderman Chart)

- 절차 설계 기법
- 논리의 기술에 중점을 둔 도형을 이용한 표현 방법으로 박스 다이어그램이라고 함
- 순차(Sequence), 선택 및 다중 선택(If ~ then ~ else, Case), 반복(Repeat ~ until, While, for) 등의 제어 논리 구조를 표현

07-기-2 **❖ 구현**

-설계단계에서 생성된 내용을 컴퓨터가 알 수 있는 형태로 변환하는 과정 (코딩)

❖ 프로그램 언어 선택 기준

-대상 업무 성격, 개발 담당자의 경험과 지식, 과거의 개발 실적 등, 4세대 언어 여부 (X)

07-기-3 **❖ 화이트 박스 테스트 : 구조 테스트**

- 모듈 안의 작동을 자세히 관찰할 수 있으며, 프로그램 원시 코드의 논리적인 구조를 커버하도록 테스트 케이스를 설계하는 프로그램 테스트 방법
- 프로그램의 제어 구조에 따라 선택, 반복 등의 부분들을 수행함으로써 논리적 경로를 제어
- 모듈 안의 작동을 직접 관찰
- 종류
 - 기초 경로 검사(Basic Path Testing, McCabe 제안), 조건 검사(Condition Testing), 루프 검사(Loop Testing), 데이터 흐름 검사(Data Flow Testing)

07-기-1 **❖ 블랙 박스 테스트 : 기능 테스트**

07-기-3

-소프트웨어가 수행할 특정 기능을 알기 위해서 각 기능이 완전히 작동되는 것을 입증하기 위한 검사

-발견할 수 있는 오류 : 성능, 부정확한 기능, 인터페이스 오류, 논리 구조상의 오류 (X)

07-기-2 종류

- 동치분할검사(Equivalence Partitioning), 경계값 분석(Boundary Value Analysis), 원인-효과 그래프 검사(Cause-Effect Graphing Testing), 오류예측검사(Fault Based Testing), 비교검사(Comparison Testing)

❖ 유지 보수

-가장 많은 비용이 투입되는 단계로써 인수, 설치된 후 발생하는 모든 공학적 작업

07-기-3 **❖ 유지보수 활동**

- 수정 보수 (Corrective) : 오류 수정
- 적응 보수 (Adaptive) : 환경 변화(하드웨어, 운영체제 등) 반영
- 완전화 보수 (Perfective) : 기능 개선, 가장 큰 비중 차지(Win98 -> Win 2000 -> Win XP)
- 예방 보수 (Preventive)

❖ 객체지향 관련 용어

-객체 (Object)

-현실 세계의 개체며 객체들 간의 상호작용은 메시지를 통해 이루어짐

-데이터 : 객체가 가지고 있는 상태 (속성, Attribute, 변수, 자료구조)

07-기-3

07-기-1

연산자 : 객체의 데이터를 처리하는 행위 (메소드, Method, 동작, Operation, 함수, 프로시저)

07-기-2

클래스 (Class)

-하나 이상의 유사한 객체들을 묶어 공통된 특성을 표현한 데이터 추상화를 의미

07-기-1

캡슐화 (Encapsulation)

07-기-2

-자료 부분과 연산(또는 함수) 부분 등 정보처리에 필요한 기능을 한 테두리로 묶는 것

07-기-3

- 07-기-1 ❖ **Rumbaugh(럼바우) 분석 기법**
 - 07-기-2 -절차 : 객체(object)모형 -> 동적(dynamic)모형 -> 기능(functional)모형
 - 07-기-3 -객체 모델링 : 시스템에서 요구되는 객체를 찾아내어 객체들의 특성을 규명
 - 동적 모델링 : 상태를 이용하여 객체 모델링에서 규명된 객체들의 행위와 상태를 시간의 흐름에 따라 표현
 - 기능 모델링 : 자료흐름도(DFD)를 이용하여 각 객체에서 수행되는 동작들을 기술
- 07-기-1 ❖ **소프트웨어 재사용**
 - 07-기-3 -이미 개발된 소프트웨어 전체 혹은 일부분을 다른 소프트웨어 개발이나 유지에 이용
 - 재사용 부품의 크기가 작을수록 재사용률이 높음
 - 클래스, 객체 등의 소프트웨어 요소는 소프트웨어 재사용성을 크게 향상시킴
 - 소프트웨어 재사용에 가장 많이 이용되는 것 : 소스 코드
 - 07-기-2 이점 : 개발시간과 비용의 단축, 소프트웨어 품질 향상, 생산성 향상, 시스템 구축방법에 대한 지식 공유, 시스템 명세, 설계, 코드 등의 문서 공유
 - 07-기-1 -재사용 라이브러리가 가져야 할 속성
 - 확장성
 - 재사용 요소들의 생성, 편집 등을 허용하는 연산
 - 편리한 접근, 탐색, 버전관리, 제어 변경
- 07-기-1 ❖ **소프트웨어 재공학 (Reengineering)**
 - 07-기-2 -기존 시스템을 이용하여 보다 나은 시스템을 구축하고 새로운 기능을 추가하여 소프트웨어 성능을 향상시키는 것

- 07-기-3 ❖ **CASE (Computer Aided Software Engineering)**
 - 소프트웨어 개발 과정에서 사용되는 요구 분석, 설계, 구현, 검사 및 디버깅 과정을 컴퓨터와 전용의 소프트웨어 도구를 사용하여 자동화하는 작업
 - 특징
 - 소프트웨어 생명 주기의 전체 단계를 연결시켜 주고 자동화시켜 주는 통합된 도구를 제공해 주는 기술
 - 개발 비용 절약, 품질 향상, 개발 기간 감소, 유지보수 간편
 - 모듈의 재사용성이 향상
 - 도구와 방법론의 결합
 - 개발 단계의 표준화
 - CASE의 정보 저장소 : 개발 과정 동안에 모아진 정보를 보관하여 관리하는 곳 -> 유지보수 용이 -> 시스템 표준화와 소프트웨어 재사용성의 기본
 - 분류
 - 상위 CASE : 요구분석과 설계 단계 지원
 - 하위 CASE : 코드 작성하고 테스트하며 문서화하는 과정 지원
 - 통합 CASE : 개발 주기 전체 과정 지원
- ❖ **상식으로 푸는 문제가 많다?**
 - 소프트웨어의 문서(document) 표준이 되었을 때, 개발자가 얻는 이득으로 가장 거리가 먼 것은?
 - 가. 시스템 개발을 위한 분석과 설계가 용이하다.
 - 나. 프로그램 유지보수가 용이하다.
 - 다. 프로그램의 확장성이 있다.
 - 라. 프로그램 개발 인력이 감소된다.

❖S/W (시스템) 개발 절차

-요구 분석 -> 설계 -> 구현(코딩) -> 테스트(시험) -> 유지보수

❖요구 분석

-어떻게 만들어 줄까? (무엇) -> 분석 도구 (ex. 개체 관계도 : ERD)

07-산-2

❖설계

-요구 분석 결과를 가지고 구체적인 기능과 구조를 체계화 (어떻게)-> 코드 설계, 입출력 설계, 파일 설계, 프로세스 설계, 프로그램 설계

❖구현

-프로그램 언어를 선정하고, 설계 명세서를 컴퓨터가 이해할 수 있도록 표현

❖테스트

-요구 사항에 맞게 작동하는가?

❖유지보수

-버전 업데이트 및 새로운 기능 추가

❖시스템의 정의

-어떤 목적을 위하여 하나 이상의 상호 관련된 요소의 유기적인 결합체

07-산-1

❖시스템의 특성

07-산-2

-종합성 : 시스템은 종합적인 결합체이다.

07-산-3

-목적성 : 시스템은 공통의 목적이 있다.

-자동성 : 시스템은 자동 조차한다.

-제어성 : 시스템은 오류 없이 처리되도록 제어한다.

07-산-1

❖피드백 (FeedBack)

-처리 결과를 평가하여 불충분한 경우, 목적 달성을 위해 반복 처리하는 요소

07-산-1

❖시스템 분석가 조건

- 기업의 목적 이해
- 업계 동향 및 관계 법규 등도 파악해야 함
- 컴퓨터 기술과 관리 기법 알아야 함
- 현장 분석 경험 중요
- 시간 배정과 계획 등을 빠른 시간 내에 파악해야 함
- 분석가는 기계 중심적이어야 함 (X) -> 인간 중심적 분석

07-산-3

❖시스템 개발 주기(순서)

-목적 설정 -> 조사 및 분석 -> 설계 -> 구현(프로그래밍) -> 테스트 -> 운용 -> 유지보수

❖코드 (Code)

-컴퓨터에서 자료처리를 쉽게 하기 위해 사용하는 기호

07-산-1

❖코드의 기능

- 3대 기능 : 배열, 분류, 식별 (정렬 X)
- 기타 기능 : 표준화, 암호화, 연상(표의성), 단순화, 오류검출, 구별 (호환 X)

07-산-2 ❖ **코드 설계 순서**

- 코드화 대상 결정
- 코드화 목적의 명확화
- 코드 부여 대상 수 확인
- 사용 범위 결정
- 사용 기간 결정
- 코드화 대상의 특성 분석
- 코드 부여 방식의 결정
- 코드의 문서화 (코드표)

❖ **코드의 종류**

- 순서 코드 : 코드화 대상 항목을 어떤 일정한 배열로 일련 번호를 배당하는 코드로서 항목 수가 적고 장래에 다시 작성하는 일이 없는 항목에 적합한 코드
- 구분 코드 (Block) : 공통성이 있는 것끼리 블록으로 구분하고, 각 블록 내에서 일련 번호를 부여하는 방법

07-산-2 **그룹 분류 코드** (Group Classification) : 일정 기준에 따라 대분류, 중분류, 소분류 등으로 구분하여 일련번호를 부여하는 방법

07-산-3 **10진 분류 코드** (Decimal) : 도서 분류 코드

07-산-3 **표의 숫자**(Significant Digit) : 길이, 넓이, 부피, 무게

07-산-1 **연상 코드** (Mnemonic, 기호) : 가전제품

07-산-2

❖ **코드 오류 종류**

- 07-산-3 **필사 오류** (transcription error, 오자오류)
 - 입력 시 임의의 한자리를 잘못 기록한 경우 (34278 -> 34578)
- 07-산-1 **전위 오류** (transposition error)
 - 입력 시 좌우 자리를 바꾸어 기록하는 경우 (1996 -> 1969)
-
- 이중 오류** (double transposition error)
 - 전위오류가 2개 이상 발생하는 경우
-
- 임의 오류**(random error)
 - 오류가 2가지 이상 결합하여 발생하는 경우

❖ **입력 설계**

- 정의 : 입력데이터를 어떤 매체를 이용하고 어떤 형태로 입력할 것인지를 설계

07-산-1 ❖ **입력 설계 순서**

- 07-산-3 **발생** : 입력 정보의 명칭과 목적 결정
- 수집
- 매체화 : 투입 매체와 입력 장치 선택, 매체화 기기 결정, 입력 정보의 형태 선택
- 투입
- 내용

❖ 데이터 입력방식

- 07-산-1 **집중 입력 방식** : 일정시간 동안 수집 -> 일괄 입력
 - 발생한 정보를 원시 전표 위에 기록하고 일정시간 단위로 수집하여 매체화 전문 기기에서 매체화해서 일괄 입력하는 시스템
- 07-산-2 **턴 어라운드 방식** : 지로 용지
 - 입력된 자료가 처리되어 일단 출력된 후 이용자를 거쳐 다시 입력되는 방식으로, 공과금, 보험료징수 등의 지로 용지를 처리하는데 사용되는 입력 방식
 - 분산 입력 방식 : 데이터를 발생한 장소에서 입력하는 방식
 - 직접 입력 방식 : 사람이 직접 손으로 작성하여 입력하는 방식 (ex. OMR)

❖ 출력 설계

-정의 : 컴퓨터가 처리한 결과를 어떤 매체를 통해, 어떤 형식으로 출력할 것인지를 설계

❖ 출력 설계 순서

- 내 용 : 출력할 항목 결정 (순서, 크기, 자릿수, 숫자, 영문자, 한글, 한자)
- 매 체 화

- 07-산-3 **분 배** : 전달 경로 결정 (분배 책임자, 분배 방법, 분배 주기)
 - 이 용 : 출력 정보명과 사용 목적 결정

❖ 출력 매체 -> COM 시스템 (Computer Output Microfilm)

07-산-3

07-산-1 ❖ 파일 설계 순서

- 07-산-2
 - 성 격 검토 (목적) : 적용 업무 확인 후 파일의 목적, 종류, 명칭을 결정
 - 항 목 검토
 - 특 성 조사
 - 매 체 검토 : 검토 후 저장 매체 결정
 - 편 성법 검토 : 순차, 색인 순차, 랜덤, 리스트 편성 등 결정

07-산-3 ❖ 순차 편성(SAM : Sequential Access Method) : 자기 테이프

- 파일 내의 각 레코드를 논리적 순서에 따라 물리적으로 연속된 위치에 기록한 파일
- 기억장소의 낭비가 없다.
- 삽입, 삭제, 검색이 어렵다.

07-산-2 ❖ 랜덤 편성 : 해싱 함수

- 처리하고자 하는 레코드를 주소 계산에 의하여 직접 처리

❖ 해싱 함수 관련 용어

- 해싱 함수 : 해시 테이블의 주소를 생성해 내는 함수
- 해시 테이블 : 해싱 함수에 의하여 참조되는 테이블
- 버킷(bucket) : 하나의 주소를 갖는 파일의 한 구역
- 슬롯(slot) : n개의 슬롯이 모여 하나의 버킷을 형성
- 충돌(collision) : 서로 다른 2개 이상의 레코드가 같은 주소를 갖는 현상

- 07-산-3 **시노임(Synonym)**: 같은 주소를 갖는 레코드의 집합
 - 오버플로 : 버킷 내에 기억 공간이 없는 현상
 - 특징 : 검색은 빠르지만 기억공간의 낭비 발생

07-산-2 ❖내용에 의한 파일 분류

- 마스터 파일 (Master) : 원장
 - 전표 처리에서 원장 또는 대장에 해당되는 파일로서 데이터 처리 시스템에서 중추적 역할을 담당하며 기본이 되는 데이터의 축적 파일
 - 트랜잭션 파일에 의해 갱신됨

07-산-1 ❖트랜잭션 파일 (Transaction) : 갱신

- 마스터 파일의 변경하고자 하는 내용을 검사하거나 갱신할 때 사용되는 정보로서, 일시적인 성격을 지닌 파일
- 히스토리 파일 (History) : 복구
 - 통계 처리나 파일의 자료에 잘못이 발생하였을 때 파일을 원상 복구하기 위해 사용되는 파일
 - 현재까지 변화된 정보를 포함하고 있는 기록 파일
- 트레일러 파일 (Trailer) : 마스터 파일 끝부분
 - 마스터 파일을 목적에 따라 여러 개의 파일로 나누었을 때 가장 끝부분에 해당하는 파일

07-산-1 ❖럼바우(Rumbaugh) 분석 기법의 절차

- 객체 모형 -> 동적 모형 -> 기능 모형
- 객체 모델링 : 시스템에서 요구되는 객체를 찾아내어 객체들의 특성을 규명
- 동적 모델링 : 상태를 이용하여 객체 모델링에서 규명된 객체들의 행위와 상태를 시간의 흐름에 따라 표현
- 기능 모델링 : 자료흐름도(DFD)를 이용하여 각 객체에서 수행되는 동작들을 기술

❖프로세스 설계의 정의

-입력 정보를 토대로 출력정보를 얻기까지의 처리과정을 설계하는 것

07-산-2 ❖프로세스 설계시 유의 사항

- 신뢰성과 정확성을 고려
- 시스템의 상태 및 구성 요소, 기능 등을 종합적으로 표시함
- 오류 체크 시스템도 고려
- 예외 사항의 처리 방법에 유의
- 하드웨어의 기기 구성 및 처리 능력을 고려
- 프로세스 전개의 사상을 통일할 것 -> 표준화
- 오퍼레이터의 개입을 많게 할 것 (X) -> 자동화 위배
- 사용자의 하드웨어와 프로그래밍에 관한 상식 수준을 고려함 (X)
- 각 부문별 담당자의 책임 범위를 고려함 (X)

❖오류 체크 시스템 - 입력 단계에서 체크

07-산-1 ❖유효 범위 체크 (Limit)

-입력 자료의 어떤 항목 내용이 논리적으로 정해진 범위 내에 있는가를 체크

07-산-2 ❖일괄 합계 체크 (Batch Total)

-입력 자료의 특정 항목 합계 값을 미리 계산해서 이것을 입력 정보와 함께 입력하고, 컴퓨터상에서 계산한 결과와 수동 계산 결과가 같은지를 체크

07-산-3 ❖균형 체크 (Balance)

-입력 정보의 두 가지 이상이 특정 항목의 합과 같다는 것을 알고 있을 때, 컴퓨터를 이용해서 계산한 결과와 분명히 같은 지를 체크
-경리 장부 처리시 차변, 대변의 한계값을 체크하는 데 사용하는 방법으로 대차의 균형이나 가로, 세로의 합계가 일치하는 가를 체크

❖처리(Process) 패턴

- 변환 (Conversion)
 - 입력 매체인 종이 테이프 또는 펀치 카드상의 데이터를 자기 디스크에 수록하는 처리 (매체 변환)
- 병합 (Merge)
 - 동일한 파일 형식을 가진 두 개 이상의 파일을 하나로 정리하는 처리 패턴
- 07-산-2 ❖갱신 (Update)
 - 07-산-3 -마스터 파일 안의 정보 변동에 의해 추가, 삭제, 교환을 하고 새로운 내용의 마스터 파일을 작성
- 분배 (Distribution)
 - 어떤 특정한 조건을 부여하여 조건을 만족시키는 정보와 만족시키지 못하는 정보로 분리하는 처리

07-산-1 ❖프로그램 설계서의 작성효과

- 프로그래머의 인사 이동시 결함을 방지
- 시스템의 수정, 유지보수가 간단
- 비용이 절감되어 장기 계획 수립 가능

07-산-3 ❖우수한 시스템의 판정 기준

- 시스템 능력
- 시스템 신뢰성
- 시스템 유연성
- 시스템 구축비용 (X)

07-산-1 ❖문서화의 특징

- 07-산-2 -프로그램 내용을 보기에 앞서 문서를 통하여 시스템에 대해 쉽게 이해할 수 있음
- 시스템 개발자 이외의 사람에게 쉽게 시스템을 이해시킬 수 있음
- 프로그램 내에도 문서화를 할 수 있음
- 문서도 시스템 구성요소의 하나
- 문서화는 시스템 개발 과정의 작업이라고 할 수 있음

07-산-2 ❖문서화의 목적

- 07-산-3 -개발 후 시스템 유지보수의 용이
- 시스템 개발 중 추가 변경에 따른 혼란 방지
- 시스템의 개발 요령과 순서를 표준화하여 보다 효율적인 작업 도모

❖소프트웨어 생명 주기

- 소프트웨어를 개발하기 위해 정의, 개발, 유지보수 과정을 각 단계별로 나눈 것
- 표현 형태 : 폭포수 모형, 프로토타입 모형, 나선형 모형

07-산-3 ❖구조적 분석의 효과

- 시스템을 하향식으로 분할할 수 있음
- 분석자와 사용자간의 의사 소통이 용이 (도형 중심)
- 전체 시스템을 일관성 있게 이해할 수 있다.
- 구조적 분석 기법(도구)
 - 자료의 흐름과 처리를 중심으로 하는 요구사항 분석 방법
 - 종류 : 자료 흐름도, 자료 사전, 소단위 명세서, 개체 관계도, 상태 전이도

07-산-2

07-산-2 ❖ 객체 (Object)

- 현실 세계의 개체며 객체들 간의 상호작용은 메시지를 통해 이루어짐
- 데이터 : 객체가 가지고 있는 상태 (속성, Attribute, 변수, 자료구조)
- 연산자 : 객체의 데이터를 처리하는 행위 (메소드, Method, 동작, Operation, 함수, 프로 시저)

07-산-1 ❖ 클래스 (Class)

- 하나 이상의 유사한 객체들을 묶어 공통된 특성을 표현한 데이터 추상화를 의미
- 공통된 속성과 연산을 갖는 객체의 집합 (객체의 일반적인 타입)

❖ 추상화 (Abstraction)

- 객체(클래스)의 속성 중 가장 중요한 것에만 중점을 두고 개략화시킨 것

❖ 인스턴스 (Instance)

- 클래스에 속한 각각의 객체 (객체는 클래스의 인스턴스)

❖ 메시지 (Message)

- 객체들 간에 상호작용을 하는데 사용되는 수단
- 객체에서 객체로 메시지가 전달되면 메소드(행위)를 시작함

07-산-2 ❖ 캡슐화 (Encapsulation)

- 데이터와 연산을 하나로 묶어 하나의 모듈 내에서 결합되도록 하는 것