

정보처리기사/산업기사 **필기**

데 이 터 베 이 스



주요포인트 꼭 알고 가야해!



[01강] 데이터베이스의 개념

1. 정보 시스템
2. 데이터베이스의 개념
3. 데이터베이스의 구성요소
4. 스키마
5. 3단계 스키마
6. DBMS의 개념과 장,단점
7. DBMS의 기능
8. 데이터베이스 언어
9. 데이터베이스 사용자



대한민국 대표 IT강사
조 대 호

여러분의 인생드라마는
지금 잘 진행되시나요?
그 드라마의 주인공은
바로 당신입니다.
이 세상 모든사람들은
당신을 위한 조연입니다.
당신의 드라마가
성공한 그날!
당신의 감사메시지 속에서
저의 이름이 불려지기를
기원합니다.

정보처리기사/산업기사

합격은 당연! 고득점 합격이 목표!



1. 정보시스템

JODAEHO

gisa plus

❖정보 시스템의 개요

- 정보 처리라고 하면 컴퓨터가 정보를 생성하기 위해 데이터를 처리하는 작업이라고 정의할 수 있다. 즉, 정보 처리는 단순한 계산이나 자료 처리보다 더 일반성을 나타내고 있다. 이것은 정보 처리가 단순히 숫자나 문자를 처리하는 데 국한되는 것이 아니라 정보를 표현하는 기호가 수치이건 비수치이건 어떤 형태의 심벌도 처리하는 사실 때문이다.
- 데이터베이스 시스템은 이러한 정보처리 시스템이 그 기능을 효율적으로 수행할 수 있게 하는 도구이다.

❖정보와 데이터

- 정보(Information)
 - 의미 없이 존재하는 데이터를 수집해서 사용자의 용도에 맞게 가공처리를 한 후 적절한 의사 결정을 할 수 있도록 **가공 처리된 지식**.
 - 정보는 정확성과 현재성을 가지고 있어야만 유용성을 가진다.
- 데이터(Data)
 - 현실 세계에 존재하는 **가공하지 않은 그대로의 모습**으로써, 현재로써는 특별히 의미를 부여할 수 없는 사실이나 값을 말한다.

❖정보 시스템

- 사회 활동을 하면서 **필요한 데이터를 수집, 저장해 두었다가 필요시에 적절한 가공에 의해서 사용자에게 필요한 정보를 만들어 내는 과정**을 말하며, 특히 기업의 기획, 운영 그리고 통제에 필요한 데이터를 제공하는 수단으로 사용되는 시스템을 경영정보 시스템(MIS ; Management Information System)이라고 한다.

주요포인트 꼭 알고 가야해!

정보처리기사/산업기사

합격은 당연! 고득점 합격이 목표!



[01강] 데이터베이스의 개념

Slides 2

❖ 데이터베이스의 정의

- 데이터베이스란 여러 조직에서 **공통의 데이터를 사용할 수 있도록 하기 위해서 만든 시스템**으로, 어느 한 조직에서만 사용하는 데이터가 아니라 공통의 목적 또는 각자의 목적에 따라서 함께 사용되는 데이터의 집합을 말한다.
- **공용 데이터** : 데이터베이스는 다양한 사용자들이 필요한 정보를 공동으로 사용할 목적으로 만들어져 있다는 의미이다.
- **저장 데이터** : 데이터 집단은 컴퓨터 시스템이 접근 가능한 저장장치에 모아져 있다는 의미이다.
- **운영 데이터** : 데이터베이스는 한 조직체가 유지되고 운영되는데 필요한 모든 개체 데이터와 관계데이터 집합이라는 의미이다.
- **통합 데이터** : 필요로 하는 데이터 집단에서 중복된 데이터나 군더더기를 제거하여 최적화시킨 데이터 집단이란 의미이다.

❖ 데이터베이스의 특성

- **실시간 접근성(Real-Time Accessibility)** : 사용자가 요구하는 상태에 대해서 일정한 입력 조건을 주었을 때 즉시 그 결과를 받을 수 있는 시스템이어야 한다. 보통 이런 시스템을 온라인 시스템, 또는 실시간 처리라고 한다.
- **계속적인 변화(Continuous Evolution)** : 데이터베이스에 존재하는 데이터는 고정적으로 있는 것이 아니라 사용자에게 의한 데이터의 삽입, 삭제, 변경이 항상 발생하며, 그 속에서도 항상 정확한 데이터를 유지해야 한다.
- **동시 공유(Concurrent Sharing)** : 데이터베이스는 한 개인이나 한 조직에서만 사용하는 것이 아니라 여러 사람과 여러 조직에서 동시에 사용할 수 있어야 하며, 자신이 필요한 데이터에 동시에 접근 할 수도 있어야 한다.
- **내용에 의한 참조(Content Reference)** : 데이터베이스는 사용자로 하여금 데이터의 저장 형태, 저장 방식 등에 대해서는 알지 못해도 사용자가 찾고자 하는 데이터의 일부 내용으로 전체 내용을 찾을 수 있는 방법을 제시해야 한다.

주요포인트 꼭 알고 가야해!

3. 데이터베이스의 구성요소

❖ 속성(Attribute)

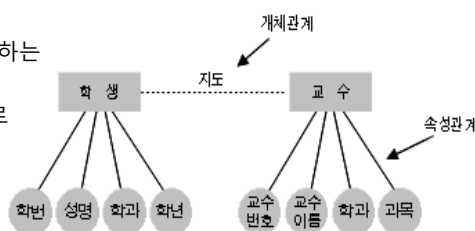
- 데이터의 가장 논리적인 단위
- 파일 구조상으로 데이터 필드(Field)에 해당된다.
- 단독으로 존재하기 어려우며, 어떤 데이터 객체(Object)의 구성요소로서 그 객체의 성질이나 상태를 기술해 주는 역할을 한다.

❖ 개체(Entity)

- 데이터베이스가 표현하려고 하는 유형, 무형의 정보 객체(Object)로서 서로 연관된 몇 개의 속성들로 구성된다.
- 파일 구조상 레코드의 개념에 해당된다.
- 단독으로 존재할 수 있으며, 정보로서의 역할을 수행한다.

❖ 관계(Relationship)

- ① 속성 관계(Attribute Relationship)
 - 개체를 기술하기 위해 그 개체가 가지고 있는 성질이나 특징을 파악하여 그 파악된 성질이나 특징을 속성으로 하여 속성과 속성 사이의 관계를 가지고 개체를 기술하게 되는데 이것을 속성 관계라고 한다.
- ② 개체 관계(Entity Relationship)
 - 개체와 개체 사이의 관계를 의미하는 것으로 실제 데이터베이스를 표현하는데 있어 자체도 데이터로 저장되어야 하므로 이것을 아주 다양하고 복잡한 문제로 나타나게 되는데 이것을 개체 관계라고 한다.

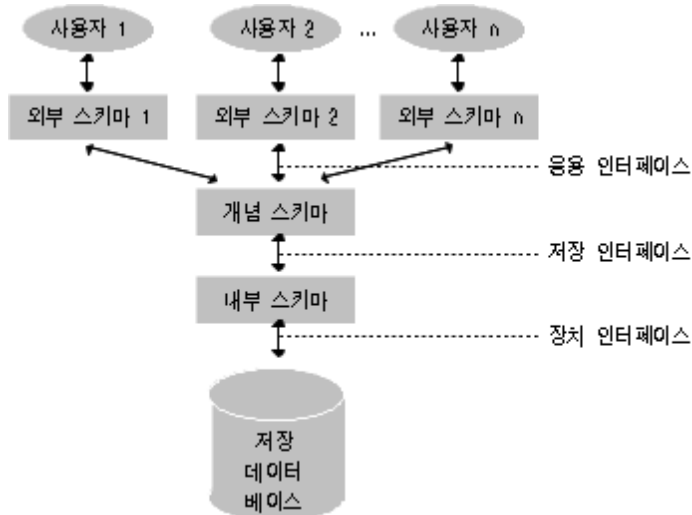


주요포인트 꼭 알고 가야해!

❖스키마의 정의

- 데이터베이스의 전체적인 논리적 설계를 의미하는 것으로 데이터베이스를 구성하는 개체, 이들의 속성, 이들 간에 존재하는 관계, 그리고 데이터의 조작 또는 이들 데이터 값이 갖는 제약조건에 관한 정의를 의미하는 것으로 외부 스키마, 개념 스키마, 내부 스키마로 나누어 기술하기 때문에 3단계 데이터베이스 구조(3-Level Database Architecture)라 한다.

❖3단계 스키마 관계도



주요포인트 꼭 알고 가야해!

5. 3단계 스키마(Schema)

❖외부 스키마(External Schema)

- 프로그램이나 일반 사용자 관점에서 개별적으로 직접 필요로 하는 데이터베이스의 논리적 구조이다.
- 전체 데이터베이스의 한 논리적인 부분을 정의한다.
- 서브스키마(subschema) 또는 뷰(View)라고도 한다.
- 각각의 사용자 그룹을 위해서 만들어져야하므로 여러 개가 존재 한다
- 응용프로그램이나 사용자들의 관심을 갖는 부분이다.

❖개념 스키마(Conceptual Schema)

- 모든 응용 시스템들이나 사용자들이 필요로 하는 데이터를 종합한 조직 전체의 데이터 베이스 구조를 논리적으로 정의하는 개념으로 하나만 존재한다.
- 범 기관적 입장에서 본 데이터베이스를 정의 한 것이다.
- 단순히 스키마라고도 한다.
- 데이터베이스 접근 권한, 보안 정책, 무결성 규정 등을 시행하는 데 필요한 요건들을 기술하고 있다.

❖내부 스키마(Internal Schema)

- 물리적 저장장치의 면에서 본 전체 데이터베이스의 구조이다.
- 개념 스키마의 저장 형태를 정의하는 개념으로 하나만 존재한다.
- 실제로 저장될 내부 레코드 형식을 정의하며, 인덱스 사용, 저장 데이터 항목의 표현방법, 그리고 내부 레코드의 물리적 순서를 기술한다.

주요포인트 꼭 알고 가야해!

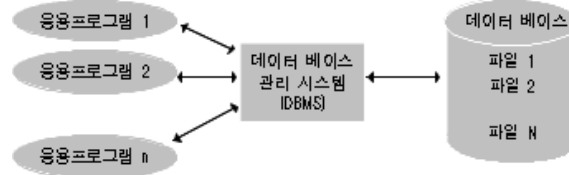
[데이터베이스의 독립성]

- 논리적 데이터 독립성
기존 응용 프로그램에 영향을 주지 않고 데이터베이스의 논리적 구조를 변경시킬 수 있는 능력을 말한다. 즉, 개념 스키마에 대한 변경을 가하면 응용 인터페이스에 대한 사상방법만 변경하면 되지 응용 프로그램이 사용하는 외부 스키마는 변경시킬 필요성이 없다는 것을 의미한다.
- 물리적 데이터 독립성
응용 프로그램이나 데이터 베이스의 논리적 구조에 영향을 주지 않고 데이터의 물리적 구조를 변경할 수 있는 것을 말한다.



❖DBMS의 정의

- 응용 프로그램과 데이터베이스의 중재자로서 모든 응용 프로그램들이 데이터베이스를 공유할 수 있게끔 관리해 주는 소프트웨어 시스템이다.
- 파일 시스템에서 야기되는 데이터 종속성과 데이터 중복성의 문제를 해결하기 위한 방법으로 제안되었다



❖DBMS의 장점

- 데이터의 일관성을 유지할 수 있다.
- 데이터의 무결성을 유지할 수 있다.
- 데이터의 중복성을 최소화한다.
- 데이터 공유를 할 수 있다.
- 표준화가 가능하다.
- 전체 데이터 요구의 조정
 - 데이터의 보안이 보장될 수 있다.

❖단점

- 운영비의 오버헤드가 발생한다.
- 자료처리 방법이 복잡해 질 수 있다.
- 복잡한 예비(Backup)와 회복(Recovery)기법이 더 어려워진다.
- 시스템의 취약성이 존재한다.

주요포인트 꼭 알고 가야해!

[데이터 종속성]

응용 프로그램과 데이터간의 상호의존관계 데이터의 구성방법이나 접근방법을 변경할 때 이것을 기초로 한 응용프로그램을 같이 변경해야한다.

[데이터 중복성]

한 시스템 내에 같은 내용의 데이터가 중복되어 저장, 관리되는 것을 말한다. 파일 시스템에서는 똑같은 데이터를 필요로 하는 경우이거나 일부분만 같은 데이터를 요구하는 경우에도 구조만 같으면 중복된 파일을 유지 된다



7. DBMS의 기능

❖DBMS의 필수 기능

- 정의 기능(Definition Facility)
 - 여러 사용자의 관점을 만족시키기 위해 데이터베이스 구조를 정의하는 기능
 - 응용 프로그램과 데이터베이스간의 상호작용 수단을 제공한다.
 - 정의 기능의 요건
 - ㉠ 데이터 모델 기술(데이터베이스의 논리적인 구조 기술)
 - ㉡ 물리적 저장에 관한 명세
 - ㉢ 물리적/논리적 사상 명세
- 조작 기능(Manipulation Facility)
 - 사용자와 데이터베이스 사이의 인터페이스 수단을 제공하는 기능
 - 사용자는 자신이 사용하는 데이터베이스에 대해서 삽입, 삭제, 변경, 검색 등의 연산을 해야 하며, 이것을 지원하는 도구가 있어야 한다.
- 제어 기능(Control Facility)
 - 데이터베이스를 관리하기 위한 제어 기능이 필수적이며, 이는 데이터의 정확성과 안정성을 보장한다.
 - 제어 기능의 3가지 요건
 - 데이터의 무결성(Integrity) 유지
 - 보안 유지 및 권한 검사
 - 데이터의 정확성 유지를 위한 병행 제어(Concurrency Control)

주요포인트 꼭 알고 가야해!

❖ 데이터 정의어(DDL : Data Definition Language)

- 사용할 데이터베이스의 정의 및 변경을 위해서 사용되는 언어
- 데이터베이스의 구조를 생성하거나 수정, 삭제하기 위해 사용되는 언어
 - 결과는 메타 데이터 형태로 데이터 사전에 기록된다.

❖ 데이터 조작어(DML : Data Manipulation Language)

- 구축된 데이터를 액세스하거나 조작하도록 하는 언어이다.
- 데이터 처리를 위해서 응용 프로그램과 DBMS 사이의 인터페이스를 제공한다.
- 데이터의 검색, 삽입, 삭제, 변경 연산 등이 있다.
- 절차적(Procedural) 데이터 조작어 : 데이터 부속어
 - 응용 프로그래머가 주로 사용하는 형태
 - 무슨 데이터를 원하며 해당 데이터에 어떻게 접근하는지를 명세 한다.
 - 한번에 하나의 레코드씩 접근하여 처리한다.
 - 독자적으로 존재하는 것이 아니라 호스트언어로 개발된 프로그램 속에 삽입(Embedded)되어 있으므로 DML 예비 컴파일러가 필요하다.
- 비절차적(Non-Procedural) 데이터 조작어(DML) : 질의어(Query)
 - 주로 터미널에서 일반 사용자가 쉽게 대화식으로 사용하는 간단한 형식의 언어
 - 독자적(Stand-alone)이고, 완전한 자료 처리 기능을 갖춘 언어
 - 데이터 접근방법을 제외한 필요한 데이터에 대한 명세 하는 언어
 - 검색, 삽입, 삭제, 갱신 연산이 가능한 대화식 고급 명령어인 질의어(Query Language) 형태이다.

❖ 데이터 제어어(DCL : Data Control Language)

- 데이터베이스는 여러 명이 동시에 사용할 수 있으므로, 함께 사용함으로써 공통의 이용 방법에 대해서 처리하는 언어

주요포인트 꼭 알고 가야해!

[데이터 정의어의 구성요소]

- ㉠ 논리적/물리적 데이터 구조 정의
- ㉡ 논리적 데이터 구조와 물리적 데이터 구조간의 사상 정의
- ㉢ DDL 컴파일러가 컴파일 후 데이터 사전에 저장

[데이터 제어어의 기능]

- ㉠ 데이터 보안(Security)
- ㉡ 데이터 무결성(Integrity)
- ㉢ 데이터 회복(Recovery)
- ㉣ 병행 수행(Concurrency)



❖ 사용자(User)

- 일반 사용자
 - 질의어를 사용해서 관리자가 정한 데이터베이스를 사용할 수 있다.
- 응용 프로그래머(Application Programmer)
 - 프로그래밍 언어(COBOL, C)에 데이터 조작어(DML)를 사용해서 데이터베이스를 사용한다.
- 데이터베이스 관리자(DBA)
 - 데이터 정의어(DDL)을 사용하여 데이터베이스를 DBMS에 표현하고 관리하는 목적으로 데이터베이스에 접근하는 사람.

❖ 데이터베이스 관리자(DBA : DataBase Administrator)

- (1) 데이터베이스 관리자(DBA)의 정의
 - DBA는 데이터베이스 시스템을 원활히 수행하도록 데이터베이스의 전체적인 관리 운영에 대한 책임을 지는 개인 또는 집단이다.
- (2) 데이터베이스 관리자(DBA)의 역할
 - 데이터베이스의 구성 요소를 결정
 - 스키마를 정의
 - 저장 구조와 접근 방법 선정
 - 보안 및 권한 부여정책, 데이터의 유효성 검사 방법 수립
 - 예비(Backup)조치, 회복(Recovery) 절차를 수립
 - 무결성 유지를 위한 대책 수립
 - 데이터 사전의 조직, 액세스, 유지관리
 - 성능 향상과 새로운 요구에 대응한 재구성
 - 데이터의 표현 표준을 설정한다.
 - 사용자의 요구 및 불평을 해결한다.
 - 시스템 이용률, 병목 현상, 성능 등 시스템을 감시한다.
 - 사용자의 요구의 변화, 데이터 사용 경향 및 각종 통계 등을 분석한다.

주요포인트 꼭 알고 가야해!

주요포인트 꼭 알고 가야해!



[02강] 데이터베이스 설계

1. 데이터베이스 생명주기
2. 데이터베이스 설계단계
3. 데이터베이스 설계시 고려사항
4. 이상현상과 정규화
5. 정규화 과정
6. 데이터베이스 모델링
7. 개체-관계(E-R)모델
8. 논리적 데이터베이스 모델링의 정의
9. 논리적 데이터베이스 모델링의 종류



대한민국 대표 IT강사
조 대 호

여러분의 인생드라마는
지금 잘 진행되시나요?
그 드라마의 주인공은
바로 당신입니다.
이 세상 모든사람들은
당신을 위한 조연이랍니다.
당신의 드라마가
성공한 그날!
당신의 감사메시지 속에서
저의 이름이 불려지기를
기원합니다.

정보처리기사/신입기사 합격은 당연! 고득점 합격이 목표! (GO)

1. 데이터베이스 생명주기

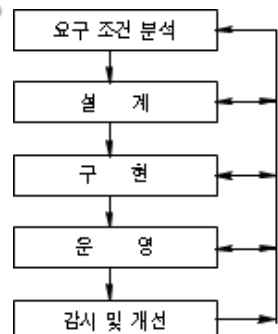
JODAEHO

gisa plus

❖ 데이터베이스 생명주기

- 정보 시스템의 기초가 되고 있는 것은 데이터베이스이다. 이 데이터베이스의 생명주기 (Life Cycle)는 다음과 같이 요구 조건 분석, 설계, 구현, 운영, 감시 및 개선 단계로 나누어 볼 수 있다.
- 요구 조건 분석 단계
 - 데이터베이스의 범위를 정의하기 위하여 사용자와 그 응용을 식별하여 그들이 필요로 하는 요구 사항을 분석하는 것이다.
- 설계
 - 데이터베이스를 설계한다. 이 설계단계는 개념적인 설계에서 시작하여 논리적인 설계를 거쳐 목표 DBMS에 구현할 수 있도록 물리적 설계까지를 모두 포함한다.
- 구현
 - 설계 다음에는 DBMS에 설계된 데이터베이스 스키마의 정의, 빈 데이터베이스 파일의 생성, 응용 소프트웨어를 구현한다. 이 단계에서는 또한 데이터를 직접 적재하거나 기존 데이터 파일을 변환하여 적재함으로써 데이터베이스를 구축한다. 또한 기존의 응용 소프트웨어가 있을 때에는 그것을 변환하여 사용할 수 있도록 한다.
- 운영
- 감시 및 개선
 - 마지막 단계인 감시 및 개선 단계는 데이터베이스 시스템을 운영하는 과정에서 발생하는 새로운 요구 조건이나 응용에 대처하고, 또한 시스템 활용의 변동에 따라 떨어질지 모르는 성능을 향상시킨다.

주요포인트 꼭 알고 가야해!



정보처리기사/신입기사 합격은 당연! 고득점 합격이 목표!



[02강] 데이터베이스 설계

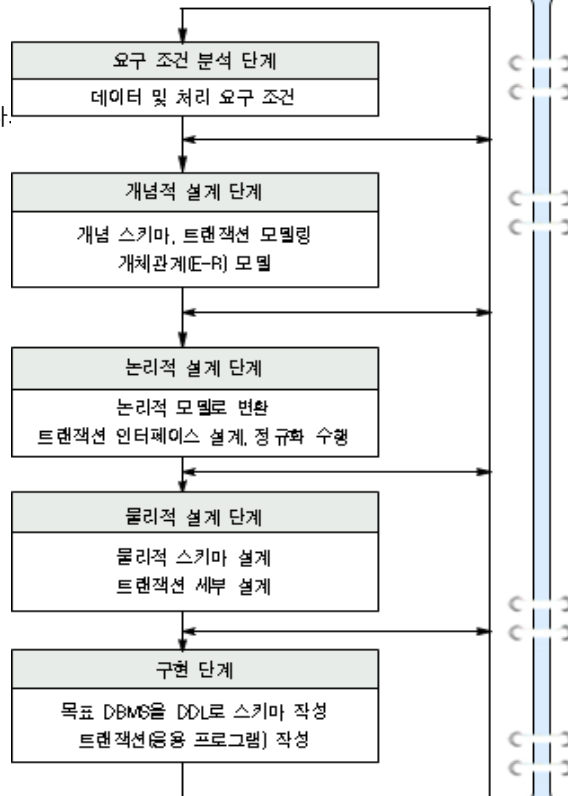
Slides 12

❖ 데이터베이스 설계

- 데이터베이스 설계는 사용자의
- 요구 조건에서부터 데이터베이스
- 구조를 도출해 내는 과정을 의미한다.

❖ 데이터베이스 설계 단계

- ㉠ 요구조건 분석
- ㉡ 개념적 설계
- ㉢ 논리적 설계
- ㉣ 물리적 설계
- ㉤ 구현



주요포인트 꼭 알고 가야해!

❖ 데이터베이스 설계 과정에서 고려해야 할 주요 사항

- 무결성(Integrity)
 - 데이터베이스에 저장된 데이터 값이 만족해야 될 제약 조건이 있다. 특히 갱신, 삽입, 삭제 등의 연산을 수행한 뒤에도 데이터 값은 이 제약 조건을 항상 만족해야 한다.
- 일관성(Consistency)
 - 저장된 데이터값 사이나 특정 질의에 대한 응답에 모순성이 없는 특성을 의미한다.
- 회복(Recovery)
 - 시스템에 장애가 발생했을 때 장애 발생 직전의 일관된 데이터베이스 상태로 복구할 수 있는 능력을 말한다.
- 보안(Security)
 - 의도적이거나 우연한 불법적인 데이터의 변경, 손실, 노출에 대한 보호를 의미한다. 이것은 데이터 조작에 대한 권한이 없는 사용자의 접근을 예방하는 것이 목적이므로 접근 제어(Access Control)와 밀접한 관계를 가지고 있다. 이 접근 제어는 특별히 허가된 사용자가 사용할 수 있는 데이터베이스의 특정 부분만을 접근할 수 있도록 하는 기법이다.
- 효율성(Efficiency)
 - 응답 시간의 단축, 저장 공간의 최적화, 시스템의 생산성 등이 포함된다.
- 데이터베이스 확장
 - 시스템 운영에 영향을 주지 않으면서 새로운 데이터를 계속적으로 추가시켜 나갈 수 있는 기법을 가지고 있어야 한다.

주요포인트 꼭 알고 가야해!

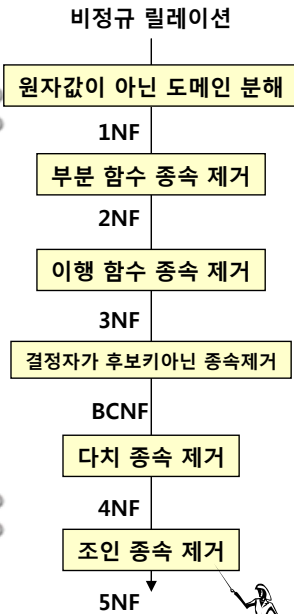
❖ 이상현상

- 관계스킴(Scheme)의 설계가 잘못되면 무엇보다도 데이터가 불필요하게 중복된다. 이 데이터 중복은 데이터 관리상의 여러 가지 치명적인 문제를 발생시킨다. 이러한 문제를 이상(Anomaly)라고 하는데 릴레이션의 데이터 값을 변경하려 할 때 이 현상이 일어나게 된다.
- 삭제 이상
 - 한 튜플(Tuple)을 삭제함으로써 유지해야 될 정보까지도 삭제되는 연쇄 삭제(Triggered deletion) 현상이 일어나게 되어 정보의 손실을 가져오게 되는 것
- 삽입 이상
 - 어떤 데이터를 삽입하려고 할 때 불필요하고 원하지 않는 데이터도 함께 삽입해야만 되는 현상
- 갱신 이상
 - 중복된 튜플들 중에서 일부 애트리뷰트 값을 갱신시킴으로써 정보의 모순성이 생기는 현상

❖ 정규화

- 이상현상이 일어나는 근본적인 이유는 여러 가지 종류의 사실들을 하나의 릴레이션으로 표현하려 하기 때문이다. 즉 애트리뷰트들 간에 존재하는 여러 가지 종속 관계를 하나의 릴레이션으로 표현하는 데서 무리가 발생하게 된 것이다. 따라서 문제의 해결은 이러한 애트리뷰트들 간의 종속성을 분석해서 기본적으로 하나의 종속성은 하나의 릴레이션으로 표현되도록 분해하면 된다. 이 분해 과정을 정규화(Normalization)라 한다.
- 정규화 이론의 기본적인 아이디어는 서로 독립적인 관계는 별개의 릴레이션으로 표현한다는 것이다. 이렇게 표현된 릴레이션이 어떤 특성의 제약 조건(constraint)을 만족하면 그 제약 조건으로 정의된 정규형(NF : Normal Form)에 속하게 된다.

주요포인트 꼭 알고 가야해!



5. 정규화 과정

■ 제 1 정규형(1NF)

- 어떤 릴레이션 R의 모든 도메인이 원자값(atomic value)만으로 되어 있다면 제 1 정규형에 속한다.

■ 제 2 정규형(2NF)

- 릴레이션 R이 제 1정규형(1NF)을 만족하면서, 키(Key)가 아닌 모든 애트리뷰트가 기본키에 완전 함수 종속이면 릴레이션 R은 제 2정규형(2NF)에 해당된다.

■ 제 3정규형(3NF)

- 릴레이션 R이 제 2정규형(2NF)을 만족하면서, 키가 아닌 모든 애트리뷰트(Attribute)들이 기본 키에 이행적 함수 종속이 아닐 때 R은 제 3정규형(3NF)에 해당된다
- 이행적 함수 종속(Transitive FD)

릴레이션 R의 애트리뷰트 A, B, C에 대해 $R.A \rightarrow R.B$, $R.B \rightarrow R.C$ 이면 $R.A \rightarrow R.C$ 이다. 이때 애트리뷰트 C는 A에 이행적 함수 종속(Transitive FD)이라고 한다.

■ 보이스 코드 정규형(BCNF ; Boyce/Code NF)

- 릴레이션 R의 모든 결정자(Determinant)가 후보 키(Candidate Key)이면 릴레이션 R은 보이스/코드 정규형에 해당된다.

■ 제 4정규형(4NF)

- 릴레이션 R에 다치 종속 $A \rightarrow B$ 가 존재할 때, 모든 애트리뷰트들 또한 A에 대하여 함수 종속이면(즉, R의 모든 애트리뷰트 X에 대해 $A \rightarrow X$ 이고 A가 후보키) 이러한 릴레이션은 제 4정규형(4NF)에 해당된다.
- 다치 종속성(Multivalued Dependency : MVD)

A, B, C 라는 3개의 애트리뷰트를 가진 릴레이션 R에서 (A, C)라는 애트리뷰트 쌍의 값에 대응하는 B의 값이 A의 값에만 종속되고 C의 값에는 독립되어 있을 때, 다치종속 $R.A \twoheadrightarrow R.B$ 가 성립한다고 한다.

■ 제 5정규형(5NF)

- 릴레이션 R이 존재하는 모든 조인 종속성(Join Dependency)이 오직 후보 키를 통해서만 성립되는 경우, 이러한 릴레이션은 제 5정규형(5NF)에 해당된다. 제 5정규형은 PJ/NF(Projection Join Normal Form)이라고 부르기도 한다.

주요포인트 꼭 알고 가야해!

[함수 종속]

어떤 릴레이션 R에서, 애트리뷰트 X의 값 각각에 대해 애트리뷰트 Y의 값이 오직 하나만 연관되어 있을 때 Y는 X에 함수 종속이라 하고, $R.X \rightarrow R.Y$ 로 표기한다.

[보이스 코드 정규화를 해야 하는 경우]

- 후보키가 여러 개일 경우
- 후보키가 복합 애트리뷰트로 구성되어 있을 경우 적용
- 후보키가 서로 중첩되었을 경우 적용

❖ 데이터 모델의 개념

- 현실 세계를 데이터베이스에 표현하는 중간 과정, 즉 데이터베이스 설계 과정에서 데이터의 구조를 논리적으로 표현하는 위해 사용되는 도구를 말하며, 데이터 모델은 현실 세계를 데이터베이스로 표현하기 위해서 적어도 개념적 구조와 논리적 구조를 거쳐 실제 데이터를 저장할 수 있는 물리적 구조로 변환하여야 한다.

❖ 개념적 데이터 모델

- 데이터 필드로 기술된 개체 타입과 이 개체 타입들 간의 관계를 이용하여 현실 세계를 표현하는 방법.
- Ex) 개체-관계 모델(E-R 모델)

❖ 논리적 데이터 모델

- 속성들로 기술된 데이터 타입과 이 데이터 타입들 간의 관계를 이용하여 현실세계를 표현하는 방법.
- Ex) 관계형 모델, 계층형 모델, 망형 모델

❖ 데이터 모델의 구성 요소

- 구조(Structure)
 - 데이터의 구조를 표현하는 것으로 데이터베이스에 표현될 대상으로서의 개체타입과 이들 간의 관계를 명세한 것
- 연산(Operation)
 - 데이터베이스에 표현된 개체 인스턴스를 처리하는 작업에 대하여 명세한 것
- 제약조건(Constraint)
 - 데이터베이스에 허용될 수 있는 개체 인스턴스에 대한 논리적 제약을 명세한 것

주요포인트 꼭 알고 가야해!

[물리적 데이터 모델]
구현을 위한 구조로 저장장치에 표현하는 형태

[데이터 모델(D)의 형태]
D = <S, O, C>



7. 개체-관계(E-R) 모델

❖ 개체-관계(E-R) 모델

- 현실 세계의 개념적 표현으로서, 개체 집합과 관계 집합을 기본 개념으로 현실 세계를 개념적으로 표현하는 방법으로, 1976년 Peter Chen이 제안하였다.
- E-R 모델의 그래픽 표현
 - : 개체 집합(Entity Set)
 - : 속성(Attribute)
 - ◇ : 개체 집합간의 관계
 - : 연결
- 관계의 유형
 - 관계(Relationship)는 개체 집합 구성 원소 사이의 대응성(Correspondence Mapping)을 명시한다.
 - 1:1 → 예) 부부관계
 - 1:n → 예) 교수-학생 관계, 사람-도시의 거주관계
 - n:m → 예) 학생-과목 관계
- 최소, 최대 대응수의 개념
 - 최소 대응수
 - 관계에 참여하는 최소 개의 개체 어커런스를 의미하는 것으로 0이면 참가해도 되고 안 해도 되는 개념이고, 1이면 최소 하나는 참가해야 된다는 의미이다.
 - 최대 대응수
 - 관계에 참여하는 최대 개의 개체 어커런스를 의미하는 것으로 1이면 최대 한 개 까지만 관계에 참여 할 수 있다는 의미이고, n이면 여러 개의 관계에 참여 할 수 있다는 의미이다.

주요포인트 꼭 알고 가야해!

❖논리적 데이터 모델의 정의

- 논리적 단계에서 사용되는 도구로서 개념적 단계에서 만들어진 모델을 컴퓨터에 저장시킬 수 있는 구조로 변환하고자 쓰이는 모델로, 데이터베이스 전체적인 논리적 구조를 설명할 때 사용되는 모델이다.

❖관계형 데이터 모델

- 데이터와 데이터간의 관계가 테이블(Table)로 표현된다.
- 데이터간의 관계를 기본키(primary key)와 이를 참조하는 외래키(foreign key)로 표현한다.
- 구조가 간단해서 이해하기 쉬우며, 데이터 조작면에서도 명확하다.
- DBMS : System R, ORACLE, SQL/DS 등

<학생>

학번	성명	학년	학과
1001	최바우	4	전산
1002	손수미	3	전자
1003	김진미	4	통계
1004	송진광	2	전산

<교과목>

과목번호	과목명	전산학점
C-101	운영체제	3
C-102	데이터베이스	3
C-103	컴퓨터구조	2

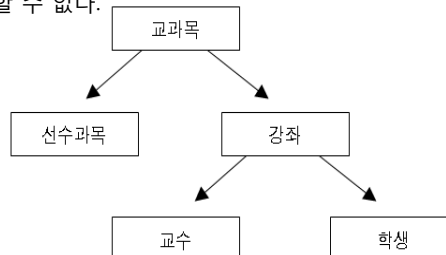
<수강>

학번	과목코드	성적
1001	C-102	A
1003	C-101	B
1001	C-103	A
1002	C-101	C
1003	C-102	C

9. 논리적 데이터 모델의 종류

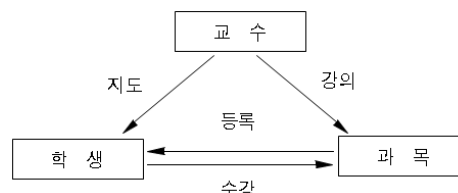
❖계층형 데이터 모델

- 데이터 간의 관계를 순서 트리 형태로 나타낸다.
- 속성 간의 관계를 레코드로, 레코드간의 관계는 링크(Link)로 표현되며, 트리(Tree) 형태로 조직된다.
- 모든 레코드 사이에는 1:n의 부모-자식(parent-child) 관계를 갖는다.
- 레코드 형태간의 n:m 링크를 직접 표현할 수 없다.
- DBMS : IMS, SYSTEM 2000 등



❖네트워크 데이터 모델

- 속성 간의 관계를 레코드로, 레코드간의 관계는 링크(Link)로 표현되며, 일반 그래프(Graph) 성질을 갖는다.
- 링크로 연결된 레코드 타입들은 오너-멤버 관계로서 오너 레코드와 멤버 레코드 사이에는 1:n의 관계를 갖는다.
- 레코드 형태간의 n:m 링크를 직접 표현할 수 없으나, 중간 레코드 형태와 두 개의 1:n 링크로 간접적으로 나타낼 수 있다.
- DBMS : DBTG, IDMS, TOTAL 등



주요포인트 꼭 알고 가야해!

[객체지향 데이터 모델]
상위 수준 구현 데이터 모델로 구분한다. 객체, 객체의 특성, 객체의 연산 등을 사용하여 데이터 베이스를 정의한다. 데이터간의 관계는 객체 식별자를 참조하는 것으로 표현한다.



주요포인트 꼭 알고 가야해!

[03강] 관계DB와 연산

1. 관계데이터베이스
2. 관계데이터베이스의 제약조건
3. 관계데이터 연산
4. 관계데이터 기본연산
5. 관계데이터 복합연산



대한민국 대표 IT강사
조 대 호

여러분의 인생드라마는
지금 잘 진행되시나요?
그 드라마의 주인공은
바로 당신입니다.
이 세상 모든사람들은
당신을 위한 조연입니다.
당신의 드라마가
성공한 그날!
당신의 감사메시지 속에서
저의 이름이 불려지기를
기원합니다.

정보처리기사/신입기사 합격은 당연! 고득점 합격이 목표! (GO)

1. 관계 데이터베이스

JODAEHO

gisa plus

❖ 관계 데이터베이스 용어

- 릴레이션 스키마(Scheme) : 한 릴레이션의 논리적 구조를 기술
- 릴레이션 인스턴스(Instance) : 어느 한 시점에 릴레이션의 내용 즉 튜플 전체를 말한다.
- 애트리뷰트(Attribute) : 관계 데이터 모델에서 데이터의 가장 작은 논리적 단위, 속성 값은 더 이상 분해하려 해도 분해할 수 없는 원자값(Atomic value)만을 갖는다.
- 튜플(Tuple) : 관계 데이터베이스 모델에서 테이블의 한 행(Row)을 구성하는 애트리뷰트의 값들의 집합. 물리적인 용어로는 레코드(record)와 같다.
- 차수(Degree) : 애트리뷰트(또는 도메인)의 개수
- 카디널리티(Cardinality) : 한 릴레이션에 포함되어 있는 튜플의 수
- 도메인(Domain) : 하나의 애트리뷰트가 가질 수 있는 같은 타입의 모든 원자값들의 집합

❖ 릴레이션의 특성

- 튜플의 유일성
 - 한 릴레이션에 포함된 튜플들은 모두 상이하다. 즉, 두 개의 똑같은 튜플은 한 릴레이션에 포함될 수 없다.
- 튜플의 무순서
 - 한 릴레이션에 포함된 튜플 사이에는 순서가 없다.
- 애트리뷰트의 무순서
 - 한 릴레이션을 구성하는 애트리뷰트 사이에는 순서가 없다.
- 애트리뷰트의 원자값
 - 모든 애트리뷰트값은 원자값(Atomic value)이다. 이 성질이 뜻하는 것은 더 이상 분해할 수 없는 원자값, 즉 처리상의 단위값(Unit value)이라는 것이다.

주요포인트 꼭 알고 가야해!

학번	성명	학과코드	학년	주민번호
100	조대호	A	1	1111
200	홍길동	B	2	2222
300	신윤복	A	3	3333
	김홍도	C	1	4444
500	김태희	F	4	5555

학과코드	학과명	담당교수
A	전자	이승엽
B	기계	박지성
C	전산	김연아



❖ 관계 데이터베이스의 키(Key)

- 후보 키(Candidate key)
 - 어떤 테이블이 주어졌을 때 그 테이블의 한 레코드를 유일하게 식별할 수 있는 키(key)의 성질을 만족하는 애트리뷰트 또는 애트리뷰트의 조합. 한 테이블에 그러한 성질을 가진 애트리뷰트가 여러 개 있을 때는 그들을 모두 후보 키라 한다.
- 기본 키(Primary Key)
 - 후보 키가 둘 이상이 되는 경우에 이 중 하나를 선정해서 주로 사용할 때 이것을 기본 키라 하고 나머지 후보키를 대체 키(Alternate Key)라 한다.
- 외래 키(Foreign Key)
 - 릴레이션 R1에 속한 한 애트리뷰트의 집합을 K라 할 때, 이 K의 값이 어떤 릴레이션 R2의 기본키로 반드시 나타나 있는 값이라면 이 K를 릴레이션 R1의 외래 키라 한다.
- 외래 키와 기본 키가 정의되는 도메인은 같아야 한다.

❖ 관계 데이터베이스의 무결성 제약(Integrity constraint)

- 개체 무결성(Entity integrity)
 - 기본 키에 속해 있는 애트리뷰트는 언제 어느 때고 널 값을 가질 수 없다.
- 참조 무결성(Referential integrity)
 - 릴레이션은 대응이 안 되는 외래키 값을 가질 수 없다. 즉 외래키 값은 널이거나 참조 릴레이션의 기본키에 반드시 존재하는 값이어야 한다.
- 영역 무결성(Domain integrity)
 - 주어진 애트리뷰트로 입력되는 모든 값은 그 애트리뷰트의 선언된 도메인 영역에 있어야 한다.

❖ 널(null)값

- 아직 알려지지 않은(모르는) 값이나, 해당 없음 등의 이유로 정보부재를 나타낼 때 사용하는 특수한 데이터 값이다.
- 공백(space)이나 영(zero)과는 다르다.

주요포인트 꼭 알고 가야해!

학번	성명	학과코드	학년	주민번호
100	조대호	A	1	1111
200	홍길동	B	2	2222
300	신윤복	A	3	3333
	김홍도	C	1	4444
500	김태희	F	4	5555

학과코드	학과명	담당교수
A	전자	이승엽
B	기계	박지성
C	전산	김연아



3. 관계 데이터 연산

❖ 관계 데이터 연산의 정의

- 관계 데이터베이스의 릴레이션을 조작하기 위한 기본연산에는 관계 대수(Relational algebra)와 관계 해석(Relational calculus)이 있다. 이 연산은 사용자의 입장에서 볼 때 데이터를 처리하는 데이터 언어가 되며 관계 대수는 절차적 언어이며 관계 해석은 비절차 언어이다.

❖ 관계 대수(Relational Algebra)

- 관계 데이터베이스의 관계(relation), 즉 테이블을 조작하는 언어.
- 하나 또는 그 이상의 테이블을 피연산자로 하여 결과로 하나의 테이블을 만들어 내는 연산을 말한다.
- 원하는 정보와 그 정보를 어떻게 유도하는가를 기술하는 절차적인 방법이다.
- 관계해석과 관계대수는 표현 방법만 다를 뿐이지 표현 능력의 차이는 없다.

❖ 기본 연산자

- Selection(σ)
 - 릴레이션에서 조건에 만족하는 레코드를 분리해 내는 연산을 의미하는 것으로 하나의 릴레이션에서 수평적인 부분 집합을 취하는 것이다.

학생 테이블

이름	학과	학번	성별
신윤복	동양화	99	여
김홍도	한국화	01	여
조대호	서양화	02	여

학과=서양화(학생)

이름	학과	학번	성별
조대호	서양화	02	여

주요포인트 꼭 알고 가야해!

[관계 해석(Relational Calculus)]

원하는 정보가 무엇이라는 것만 선언적으로 정의하는 비절차적인 성격을 갖는다. 관계 해석은 원래 수학의 프레딕트 해석에 기반을 두고 있으며 튜플 관계 해석과 도메인 관계 해석이 있다.

(1) 튜플 관계 해석
원하는 정보가 무엇이지를 표현하는데 있어 기본적인 연산 단위를 튜플 단위로 해서 명시하는 방법을 의미한다.

(2) 도메인 관계 해석
원하는 정보가 무엇이지를 표현하는데 있어 기본적인 연산 단위를 도메인 단위로 해서 명시하는 방법을 의미한다.



■ Projection(Π)

- 릴레이션에서 참조하고자 하는 어트리뷰트를 선택하여 분리해내는 연산을 의미하며 하나의 릴레이션에서 수직적인 부분집합을 취하는 방법이다.
- 릴레이션 생성 과정에서 똑같은 튜플이 중복되게 생성되면 시스템은 그 중 하나만 제외하고 나머지는 모두 제거한다.

학생 테이블				Π 이름, 학과(학생)	
이름	학과	학번	성별	이름	학과
신운복	동양화	99	여	신운복	동양화
김홍도	한국화	01	여	김홍도	한국화
조대호	서양화	02	여	조대호	서양화

■ Union($\cup$)

- 관계성이 있는 두 개의 릴레이션을 합쳐 하나의 릴레이션을 만들어 내는 연산을 말한다.

■ Difference($-$)

- 두 개의 릴레이션이 있을 때 그 중에서 하나의 릴레이션에서 또 다른 릴레이션의 내용을 제거해서 새로운 릴레이션을 생성하는 연산을 말한다.

■ Cartesian product($\times$)

- 두 개의 릴레이션이 있을 때 두 릴레이션의 레코드들의 순서쌍의 집합을 만들어 새로운 릴레이션을 만들어 내는 연산을 말한다.

주요포인트 꼭 알고 가야해!

◆ 복합 연산자

■ Intersection($\cap$)

- 관계성이 있는 두개의 릴레이션에서 중복되어 있는 내용을 선택하여 새로운 릴레이션을 만들어 내는 연산을 의미한다.

■ Join($\bowtie$)

- 각 릴레이션을 합치는 조건에 따라 새로운 튜플이 만들어지는 연산
- EX> 학생[학번 $\bowtie$ 학번]신상명세서

학생 테이블과 신상명세서 테이블에서 학번이 같은 학생에 대한 두 레코드의 카티션 프로덕트

학생 테이블				신상명세서 테이블			
학번	이름	나이	학과	학번	성적	성별	주소
00	김태희	21	컴공	00	F	남	부산
01	장운정	20	컴공	01	A	남	서울
99	정지훈	22	전산	98	B	남	대구

학번	이름	나이	학과	성별	성적	주소
00	김태희	21	컴공	남	F	부산
01	장운정	20	컴공	남	A	서울

■ Division($\div$)

- 나누기 기호로 사용되며 $R(X, Y)$ 에 대해 $S(Y)$ 와 같은 레코드 중 $R(X)$ 릴레이션을 결과로 하는 연산자

- $R \div S$: R의 테이블 중에

필드가 S인 레코드를

뽑아낸 후, 뽑아낸

테이블 중에서

S의 필드를 제거한

테이블

학생 테이블		과목테이블		학생 테이블-과목 테이블	
학번	과목코드	과목코드		학번	
100	A	A		100	
100	B	C		200	
200	C			300	
300	A				

주요포인트 꼭 알고 가야해!

주요포인트 꼭 알고 가야해!



[04강] SQL

1. SQL의 정의
2. 정의문의 사용예
3. 조작성
4. 조작성 사용예 1
5. 조작성 사용예 2



대한민국 대표 IT강사
조 대 호

여러분의 인생드라마는
지금 잘 진행되시나요?
그 드라마의 주인공은
바로 당신입니다.
이 세상 모든사람들은
당신을 위한 조연이랍니다.
당신의 드라마가
성공한 그날!
당신의 감사메시지 속에서
저의 이름이 불려지기를
기원합니다.

정보처리기사/산업기사 합격은 당연! 고득점 합격이 목표! (GO)

1. SQL의 개념

JODAEHO

gisa plus

❖SQL(Structured Query Language)

- 1974년에 IBM연구소에서 개발 발표한 SEQUEL에 연유한 데이터 언어로 개발되었다.
- 관계 대수와 관계 해석을 기초로 한 혼합 언어이다.
- 데이터 정의, 데이터 조작, 그리고 제어 기능을 모두 제공하고 있다.
- 온라인 터미널을 통해 대화식 질의어로 사용할 수 있다.
- PL/1, COBOL, PASCAL, C등의 범용 프로그래밍 언어로 쓰여진 응용 프로그램에 삽입된 형태로도 사용이 가능하다.
- 레코드 집합 단위로 처리하는 언어이다.
- 데이터 처리를 위한 접근 경로에 대한 명세가 필요하지 않으므로 비절차적 언어라고 할 수 있다.

❖SQL에서 사용되는 테이블의 종류

- SQL은 기본테이블(Base Table), 뷰(View), 임시테이블(Temporary Table)을 관리한다.
- 기본 테이블은 DDL에 의해 생성되는 테이블이고 독자적으로 존재한다.
- 뷰는 DDL로 만들어지지만 임의의 기본 테이블로부터 파생되어 생성되는 가상 테이블(Virtual Table)이다.
- 임시 테이블은 DDL에 의해 만들어지는 것이 아니고 질의문 처리 과정에서 DML에 의해 만들어지는 테이블로 임시적으로 만들어지는 테이블이다.

❖데이터 정의문 기본구조

- CREATE TABLE 기본_테이블_이름
- ALTER TABLE 테이블_이름
- DROP TABLE 기본_테이블_이름 [CASCADE | RESTRICT];
- CASCADE : 참조되는 테이블이 있으면 상관없이 모두 삭제.
- RESTRICT : 참조되는 테이블이 있으면 명령 거부.

주요포인트 꼭 알고 가야해!

정보처리기사/산업기사 합격은 당연! 고득점 합격이 목표!



[04강] SQL

Slides 28

❖정의문 사용예제

▪CREATE TABLE 직원

번호 INT,
이름 CHAR(10) NOT NULL,
전화번호 CHAR(8),
성별 GENDER,
나이 INT,
봉급 DECIMAL(8),
부서번호 CHAR(4),
PRIMARY KEY (번호),
FOREIGN KEY (부서번호), REFERENCES 부서(부서번호),
CHECK (나이 >= 20 AND 나이 <= 60));

직원

번호	이름	전화번호	성별	나이	봉급	부서번호

▪ALTER TABLE 직원 ADD 주소 VARCHAR(50);

▪DROP TABLE 직원;

기본테이블이 제거되면 그 테이블로부터 생성된 인덱스나 뷰도 자동으로 제거된다.
Index문에서는 Alter문을 사용하지 못한다.
기본테이블이 변경되면 인덱스도 자동으로 변경된다.

주요포인트 꼭 알고 가야해!

부서

부서번호	부서명

3. 조작문

❖데이터 조작문종류

- SELECT(검색)
- INSERT(삽입)
- DELETE(삭제)
- UPDATE(갱신)

과목

과목 번호	과목 이름	학 점	학 과	담당 교수
C123	프로그래밍	3	컴퓨터	김성기
C312	자료 구조	3	컴퓨터	황수찬
C324	파일 처리	3	컴퓨터	이규철
C413	데이터 베이스	3	컴퓨터	이석호
E412	반도체	3	전 자	홍봉희

❖데이터 조작문 기본구조

SELECT [ALL|DISTINCT] 열_리스트
FROM 테이블_리스트
[WHERE 조건]
[GROUP BY 열_리스트]
[HAVING 조건]
[ORDER BY 열_리스트 [ASC|DESC]];

등록

학 번	과목 번호	성 적	중간 성적	기말 성적
100	C413	A	90	95
100	E412	A	95	95
200	C123	B	85	80
300	C312	A	90	95
300	C324	C	75	75
300	C413	A	95	90
400	C312	A	90	95
400	C324	A	95	95
400	C413	B	80	85
400	E412	C	65	75
500	C312	B	85	80

학생

학 번	이 름	학 년	학 과
100	나연목	4	컴퓨터
200	이찬영	3	전 기
300	정기태	1	컴퓨터
400	송병호	4	컴퓨터
500	박종화	2	산 공

주요포인트 꼭 알고 가야해!

❖SELECT 문

- ex01) 학생 테이블에서 학번과 이름을 검색하라.
SELECT 학번, 이름 FROM 학생 ;
- ex02) 학생 테이블을 검색하라.
SELECT * FROM 학생 ;
- ex03) 학생 테이블에 있는 학과들을 검색하라.
SELECT DISTINCT 학과 FROM 학생 ;
- ex04) 학생 테이블에서 학과가 '컴퓨터'이고 학년이 4인 학생의 학번과 이름을 검색하라.
SELECT 학번, 이름 FROM 학생 WHERE 학과 = '컴퓨터' AND 학년=4 ;
- ex05) 과목별 기말 성적의 평균을 검색하라.
SELECT 과목번호, AVG(기말성적) FROM 등록 GROUP BY 과목번호 ;
- ex06) 3사람 이상 등록한 과목의 성적 평균을 검색하라.
SELECT 과목번호, AVG(기말성적) FROM 등록 GROUP BY 과목번호
HAVING COUNT(*) >= 3
* HAVING구는 반드시 GROUP BY 절과 함께 사용되며, 단독으로 사용할 수 없다.
- ex07) 등록 테이블에서 중간 성적이 90점 이상인 학생에 대해 학번의 내림차순으로,
또 같은 학번에 대해서는 과목 번호의 오름차순으로 학번과 과목 번호를 검색하라.
SELECT 학번, 과목번호 FROM 등록 WHERE 중간성적 >= 90
ORDER BY 학번 DESC, 과목번호 ASC ;
- ex08) 과목번호 'C413'에 등록한 학생의 이름, 학과, 성적을 검색하라.
SELECT 학생.이름, 학생.학과, 등록.성적 FROM 학생, 등록
WHERE 학생.학번 = 등록.학번 AND 등록.과목번호='C413' ;
- ex09) 과목번호 'C413'에 등록한 학생의 이름을 검색하라.
SELECT 이름 FROM 학생
WHERE 학번 IN (SELECT 학번 FROM 등록 WHERE 과목번호='C413');

주요포인트 꼭 알고 가야해!

5. 조작성 사용에 2

- ex10) 학생수를 검색하라.
SELECT COUNT(*) FROM 학생 ;
* COUNT(*)의 처리 대상은 행(튜플)의 집합이고
그 결과는 그 집합에 속하는 행(튜플)의 수이다. 이때 중복되는 행도 함께 포함된다.
- ex11) 학번이 300인 학생이 등록한 과목은 몇 개인가?
SELECT COUNT(DISTINCT 과목번호) FROM 등록 WHERE 학번=300 ;
- ex12) 과목 번호가 C로 시작되는 과목의 번호와 과목 이름을 검색하라.
SELECT 과목번호, 과목이름 FROM 과목 WHERE 과목번호 LIKE 'C%';
* LIKE 프리디킷의 기호
'%' : 어떤 길이의 어떤 문자 스트링도 관계없다는 의미
'_' : 임의의 1문자를 대신하는 의미
부분매치 질의문은 검색속도가 느린 단점을 가진다.
- ex13) 학과가 NULL인 학생의 학번과 이름을 검색하라.
SELECT 학번, 이름 FROM 학생 WHERE 학과 IS NULL
* NULL을 검색조건에 명세할 때는
= 또는 <> 대신에 IS 또는 IS NOT을 사용해야만 한다.

❖UPDATE 문

- ex) 100번 직원의 나이를 29로 변경하라
UPDATE 직원 SET 나이 = 29 WHERE 번호 = 100;

❖INSERT 문

- ex) 직원번호 600, 이름 홍길동, 나이 27, 봉급 200, 부서 '교육부'인 직원을 삽입하라
① INSERT INTO 직원(번호, 이름, 나이, 봉급, 부서)
VALUES (600, '홍길동', 25, 100, '교육부');
② INSERT INTO 직원 VALUES (600, '홍길동', 25, 100, '교육부');

주요포인트 꼭 알고 가야해!

[DELETE 문]

- ex) 100번 직원을 삭제하라
DELETE FROM 직원
WHERE 번호 = 100;

- ex) 홍길동과 동일 부서에
근무하는 모든 직원을 삭제
하라
DELETE FROM 직원
WHERE 부서번호 =
(SELECT 부서번호
FROM 직원
WHERE 이름 = '홍길동');

주요포인트 꼭 알고 가야해!

[05강] 고급 데이터베이스

1. 삽입SQL, 시스템카탈로그
2. VIEW
3. 트랜잭션
4. 데이터베이스 제어
5. 병행제어
6. 회복
7. 분산 데이터베이스

합격
비법



대한민국 대표 IT강사
조 대 호

여러분의 인생드라마는
지금 잘 진행되시나요?
그 드라마의 주인공은
바로 당신입니다.
이 세상 모든사람들은
당신을 위한 조연이랍니다.
당신의 드라마가
성공한 그날!
당신의 감사메시지 속에서
저의 이름이 불려지기를
기원합니다.

정보처리기사/신입기사 합격은 당연! 고득점 합격이 목표! (GO)

1. 삽입SQL, 시스템카탈로그

JODAEHO

gisa plus

❖ 삽입 SQL(Embedded SQL)

- 호스트 언어에 포함되어 사용하는 SQL을 내장SQL이라하며 **데이터 조작어를 호스트 프로그래밍언어에 삽입하여 사용될 때 이 데이터 조작 명령들을 지칭해서 데이터 부속어**라 한다.
- 특성
 - 명령문 앞에 EXEC SQL을 붙여 사용
 - 호스트 언어의 실행문이 나타날 수 있는 곳이면 어디나 나타날 수 있다.
 - 호스트 언어의 변수를 포함할 수 있다.
 - 호스트 변수는 다른 SQL 필드 이름과 구별하기 위해 콜론(:)이 앞에 붙는다.
- 커서를 이용한 조작
 - SQL의 SELECT에 의해 검색되는 결과는 몇 개의 튜플로 구성되는 반면 호스트 변수는 한번에 하나의 레코드만 취급 가능하므로 **SQL의 레코드 집합 단위 처리와 호스트 언어의 개개 레코드 단위 처리사이에 인터페이스 역할이 요구되는데 이때 커서를 사용한다.**
 - 커서는 일종의 포인터로 DECLARE CURSOR문 사용한다.

❖ 시스템 카탈로그

- 데이터 사전이라고도 하며 시스템이 필요로 하는 테이블, 뷰, 인덱스, 응용계획, 접근 권한, 무결성, 제약 사항에 관한 **모든 정보를 메타 데이터 형태로 가지고 있는 시스템 데이터베이스를 의미한다.**
- 시스템 카탈로그의 특징
 - 시스템 자신이 필요로 하는 여러 가지 객체에 관한 정보를 포함하고 있는 시스템 데이터베이스로서, 포함하고 있는 객체로는 테이블, 데이터베이스, 뷰, 접근 권한 등이 있다.
 - **DBMS가 스스로 생성하고, 유지하는 데이터베이스 내의 특별한 테이블의 집합**
 - 데이터베이스 스키마에 대한 정보를 제공하며 통계정보가 저장될 수 있다.
 - 시스템 카탈로그에 대한 갱신은 DBMS가 자동적으로 수행한다.

주요포인트 꼭 알고 가야해!

[데이터 사전(Data Dictionary)]

데이터베이스 관리자(DBA)의 도구로서 시스템 내의 모든 개체들에 대한 정의나 명세에 관한 정보를 수록하며 시스템 카탈로그(Catalog)라고도 한다.

[데이터 디렉토리(Data Directory)]

데이터 사전에 수록된 데이터를 참조하는 데 필요한 정보를 수록한다. 데이터 사전은 사용자와 시스템이 모두 사용할 수 있는 반면, 데이터 디렉토리는 시스템에서만 사용할 수 있다.



정보처리기사/신입기사 합격은 당연! 고득점 합격이 목표!

(GO)

[05강] 고급 데이터베이스

Slides 34

❖SQL 뷰(View)

- 하나 또는 그 이상의 테이블로부터 유도되는 가상 테이블이다.
- 뷰는 다른 뷰를 정의하는데 기초가 될 수 있다.
- 터미널 사용자는 데이터 검색 시 복잡한 명령을 간단하게 처리한다.
- 뷰의 정의 내용에 따라 일부 연산의 제한을 받는다.
 - 저장 데이터는 제공하지 않는 대신에 다른 테이블에 의해 정의되며, 시스템 카탈로그에 저장되었다가 실행시간에 테이블을 구축하는 것이다.

❖뷰의 생성

- CREATE VIEW 뷰_이름[(열_이름 리스트)] AS SELECT 문 [WITH CHECK OPTION];

❖뷰의 제거

- 뷰의 제거는 DROP으로 제거할 수 있지만, 일반적인 ALTER문으로는 변경할 수가 없다.
- DROP VIEW 뷰_이름 { CASCADE | RESTRICT };

❖뷰의 장점

- 데이터의 논리적 독립성을 어느 정도 제공한다.
- 데이터의 접근을 제어하게 함으로써 보안을 제공한다.
- 사용자의 데이터 관리를 간단하게 해 준다.
 - 여러 사용자의 상이한 응용이나 요구를 지원해 준다.

❖뷰의 단점

- 독자적인 인덱스를 가질 수 없다.
- 그 정의를 변경할 수 없다. ALTER문 사용이 불가능
- 삽입, 삭제, 갱신 연산에 많은 제한을 가지고 있다.

주요포인트 꼭 알고 가야해!

[뷰의 조작 연산]

뷰는 SELECT문을 일반 테이블처럼 사용할 수 있으나 삽입(INSERT)이나 삭제(DELETE), 갱신(UPDATE)문을 뷰에 사용하는 것은 피해야 한다.



3. 트랜잭션

❖트랜잭션의 정의

- 데이터베이스에서 일어나는 연산의 집합이다.
- 하나의 논리적인 기능을 수행하기 위한 작업의 단위이다.
- 사용자가 제기한 데이터베이스의 작업수행을 위한 데이터 조작어(DML)의 집단이다.
- 데이터베이스의 상태를 일관적 상태로 유지하기 위한 병행수행제어(Concurrency) 및 회복(Recovery)의 기본단위이다.
- 데이터베이스의 상태를 하나의 일관된 상태로 변환시킨다.

❖트랜잭션의 특성

- (1) 원자성(Atomicity) : 트랜잭션은 완벽하게 수행 완료되거나, 전혀 수행되지 않아야 한다.(all or nothing)
- (2) 일관성(Consistency) : 트랜잭션이 그 실행을 성공적으로 완료하면 언제나 일관성 있는 데이터베이스 상태로 변환한다.
- (3) 격리성(Isolation) : 트랜잭션의 수행 중에는 다른 트랜잭션 연산을 수행할 수 없다.
- (4) 영속성(Durability) : 완료된 트랜잭션의 결과는 영구적으로 유지되어야 한다.

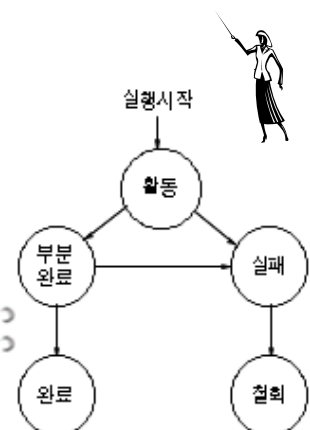
❖트랜잭션 상태

- (1) 활동(Active) : 트랜잭션이 Begin_Trans에서부터 실행을 시작하였거나 실행중인 상태
- (2) 부분 완료(Partially Committed) : 트랜잭션이 마지막 명령문을 실행한 직후의 상태
- (3) 실패(Failed) : 정상적 실행을 더 이상 계속할 수 없어서 중단한 상태
- (4) 완료 : 트랜잭션이 실행을 성공적으로 완료하여 Commit 연산을 수행한 상태
- (5) 철회 : 트랜잭션이 실행에 실패하여 Rollback 연산을 수행한 상태
- (6) 트랜잭션의 원자성을 위한 연산
 - ① Commit : 트랜잭션의 실행이 성공적으로 종료되었음을 선언하는 연산
 - ② Rollback : 트랜잭션의 실행이 실패하였음을 선언하는 연산

주요포인트 꼭 알고 가야해!

[ACID]

Atomicity
Consistency
Isolation
Durability



❖ 보안(Security)

- 뜻밖의 사고 또는 권한이 없는 사용자로부터 데이터의 누설, 변경, 파괴에서 데이터베이스를 보호하려는 것을 말한다.
- 사용자 구분과 대조 확인
- 권한 부여 방법
 - ① View를 이용한 권한 부여
 - ② Grant, Revoke 명령문을 이용한 권한 부여와 해제
[예제]
DBA : GRANT SELECT ON STUDENT TO U1 WITH GRANT OPTION;
U1 : GRANT SELECT ON STUDENT TO U2;
DBA : REVOKE SELECT ON STUDENT FROM U1 CASCADE;

❖ 데이터 암호화 기법

- 공통 키 방식(대칭형 암호화 알고리즘)
 - 네트워크 상에서 송신자가 암호키로 암호화하여 보내면 수신자가 동일한 암호키로 복호화하는 방식으로 DES, IDEA, RC2 등이 있다.
- 공개 키(Public-Key) 방식(비대칭형 암호화 알고리즘)
 - 데이터를 암호문으로 변환할 때의 암호 키와 암호문을 평문으로 변환시킬 때 복호키가 서로 다르며 암호화하는 암호키는 공개하고 복호키는 비밀로 해 데이터의 송수신시 보안을 유지하는 방법으로 RSA 기법 등이 있다.

❖ 무결성

- 데이터베이스에 있는 데이터 값의 정확성(Correctness), 정밀성(Accordancy), 유효성(Validity)을 보장하는 것으로 무결성의 핵심은 데이터베이스에 저장된 데이터 값을 정확하게 유지하는데 있다.

주요포인트 꼭 알고 가야해!

[무결성 종류]

1. 개체 무결성(Entity integrity)
2. 참조 무결성(Reference integrity)
3. 영역 무결성(Domain integrity)



5. 병행제어

❖ 병행수행 제어

- 동시에 실행되는 트랜잭션들이 데이터베이스의 일관성을 파괴하지 않도록 트랜잭션의 상호작용을 제어하는 것을 말한다.

❖ 병행 수행의 문제점

- 갱신 분실(lost update)
- 모순성(inconsistency)
- 연쇄 복귀(cascading rollback) or 회복불능(undeliverability)

❖ 병행 수행의 기법

- 병행처리가 제대로 수행되기 위해서는 모든 트랜잭션에 직렬성이 보장되어야 한다
- 직렬성은 동시에 처리한 트랜잭션들의 결과가 트랜잭션들을 임의의 순서로 처리한 결과와 같도록 보장하는 방법이다.
- 로킹(Locking) 기법
 - 병행 수행 시 문제점 해결을 위해 트랜잭션이 어떤 데이터에 대해서 액세스하는 동안에 또 다른 트랜잭션이 이 데이터에 대해 액세스하기 위해서는 LOCK을 소유하고 있는 경우에만 가능하도록 하는 방법으로 LOCK을 소유하고 해제하는 가 하는 로킹 규칙을 정해놓고 해결하는 방법이다.
- 타임 스탬프 순서(Time Stamp Ordering) 기법
 - 트랜잭션이 발생했을 때 미리 이 트랜잭션에 대해 시간에 따른 순서를 부여해서 이 순서에 따라 각 트랜잭션들이 이 데이터를 처리하도록 하는 방법이다.
- 검증 기법
- 다중 버전 기법
 - 교착상태
 - 로킹 규약은 직렬성을 보장 할 수 있으나 교착상태가 발생할 수 있다는 문제점이 있다. 이 교착상태라는 것은 두개 이상의 트랜잭션이 있을 때 이 트랜잭션들이 서로 실행이 완료되기를 무한정 기다리는 것을 의미한다

주요포인트 꼭 알고 가야해!

[로킹의 단위]

- ① 로킹 단위를 작게 했을 때 : 병행성 수준은 뛰어나지만 관리가 어렵다.
- ② 로킹 단위를 크게 했을 때 : 병행성 수준은 낮아지지만 관리가 쉽다.

[교착상태 예방법]

- ① 트랜잭션이 일을 시작하기 전에 필요한 모든 데이터에 대해 미리 로크를 요청하는 방법
- ② 처리할 항목에 대하여 부분적인 순서를 부여하여 로크를 요청하는 방법
- ③ wait-die 방법
- ④ wound-wait 방법



❖회복(Recovery)

- 여러 가지 이유의 장애로 인해 손상된 데이터베이스를 손상되기 이전의 정상적인 상태 (Consistent State)로 복구시키는 작업이다.
- 장애의 유형
 - 액션 장애(Action Failure, 논리적 오류) : 사용자가 예견할 수 있는 실패로 **응용 프로그램 내의 예러 조건을 만족하는 경우**로 DBMS의 회복 대상이 아니다.
 - 트랜잭션 장애(Transaction Failure) : **비정상적인 프로그램의 종료**로 인한 실패로 사용자가 예측할 수 없는 실패를 말한다.
 - 시스템 장애(System Failure) : 정전 등으로 인해 주메모리(데이터베이스 버퍼)의 내용이 손실된 경우로 DBMS는 종료되지 않은 트랜잭션들을 ROLLBACK 시킨다.
 - 미디어 장애(Media Failure) : 디스크 헤드 고장, 디스크 채어기 고장 등으로 디스크에 저장된 데이터베이스의 일부가 손상된 경우로 DBMS는 유틸리티 프로그램 (Dump /Restore) 등을 이용하여 데이터베이스를 회복한다.
- 회복 연산
 - REDO : 갱신이 완료된 데이터를 로그를 이용해 데이터베이스에 적용시키는 연산으로 주로 데이터베이스의 **내용이 손상 됐을 때**, BACK본으로 회복한 다음 BACK본에 있는 데이터 이후는 로그에 갱신되어 있는 데이터를 데이터베이스에 적용하는데 사용한다.
 - UNDO : 데이터베이스 내용 자체는 손상되지 않았지만 **내용에 대한 신뢰성을 잃어버려** 모든 변경을 취소시킴으로써 원래 데이터베이스로 복원하는 경우에 사용한다.
- 회복 기법
 - 로그(Loge)를 이용한 기법
 - 로그(Loge)를 이용하지 않는 기법
 - UNDO 작업은 갱신된 데이터를 이전 값으로 복구
 - 체크 포인트(Checkpoint) 기법

주요포인트 꼭 알고 가야해!

[로그(Loge)를 이용한기법]

- ① 즉시 갱신을 로그를 이용한 회복기법
- ② 로그를 이용한 지연 회복기법
- ③ 검사 시점에 의한 회복기법

[로그(Loge)를 이용하지 않는 기법]

- ① 그림자 페이징 회복기법
- ② 미디어에 의한 회복기법



7. 분산데이터베이스

❖분산 데이터베이스 개념

- 분산 데이터베이스는 전체가 동일한 지역에 저장되어 있지 않고 데이터가 여러 개 지역으로 분산되어 존재하며 고속의 통신 매개체를 이용하여 서로 연결되어 있는 다른 컴퓨터와 데이터를 주고받을 수 있다.

❖온라인 처리 시스템과 분산처리 시스템

- 온라인 처리 시스템
 - 온라인 처리 시스템은 구성 면에서 비용이 적게 들고, 메인 컴퓨터에서 데이터를 이용하기가 쉬우며 통제하기가 편리한 면이 있다. 그러나 통신비용이 많이 들고 중앙의 컴퓨터 시스템이나 통신 시설의 장애 발생시 파일에 많은 영향을 끼친다.
- 분산 처리 시스템
 - 컴퓨터 시스템 및 데이터베이스들이 지역적으로 분산되어 있으면서 정보교환을 위해 네트워크로 상호 결합되어 있는 시스템을 분산 처리 시스템이라 한다. 중앙의 호스트 컴퓨터와 각 노드들은 자체의 데이터를 가지고 있으면서 데이터 통신은 물론 처리 및 공유도 가능하다.

❖분산데이터베이스의 장점

- 데이터 공유와 분산제어가 가능하다.
- 중앙의 컴퓨터가 고장 나더라도 전체 시스템에는 영향을 끼치지 않는다.
- 데이터 통신비용이 감소된다.
- 시스템 성능이 높아진다.

❖분산데이터베이스의 단점

- 소프트웨어 개발비용이 비싸다
- 구현이 어렵다.
- 보안이 어려워진다.

주요포인트 꼭 알고 가야해!

주요포인트 꼭 알고 가야해!



[06강] 자료구조

1. 자료구조의 종류
2. 선형구조
3. 비선형구조-Tree
4. 이진 Tree
5. 트리운행법, 수식표기법
6. 비선형구조-Graph



대한민국 대표 IT강사
조 대 호

여러분의 인생드라마는
지금 잘 진행되시나요?
그 드라마의 주인공은
바로 당신입니다.
이 세상 모든사람들은
당신을 위한 조연입니다.
당신의 드라마가
성공한 그날!
당신의 감사메시지 속에서
저의 이름이 불려지기를
기원합니다.

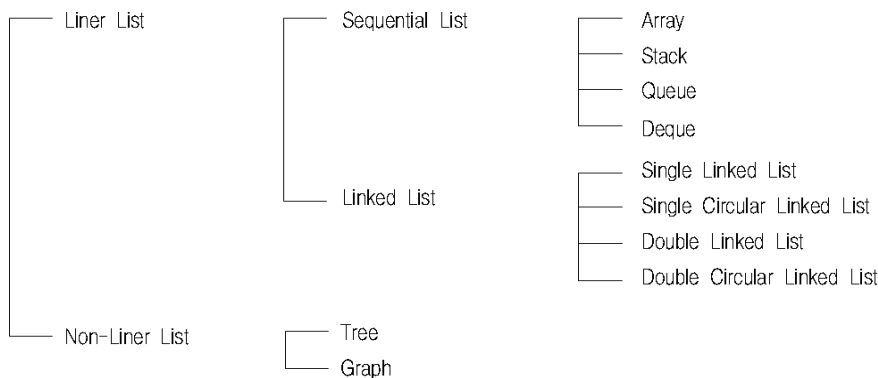
정보처리기사/산업기사 합격은 당연! 고득점 합격이 목표! (GO)

1. 자료구조의 종류

JODAEHO

gisa plus

❖자료의 표현



❖자료의 표현 단위

- Bit : Binary Digit의 약자로 2진수 0, 1 한자리를 의미, 정보표현의 최소 단위
- Nibble : 4Bit 단위의 표시
- Byte : 주소지정의 최소단위, 1 Byte = 8bit
- Word : 전산기 내부의 명령어 처리 단위
 - Half word = 2 Byte, Full word = 4 Byte, Double word = 8 Byte
- Field(=Item) : 의미를 가지는 최소단위 ex) 이름, 학년, 주소 등
- Record : Program 상에서 취급되는 자료 처리 단위, 정보처리의 기본단위
- Block : 입출력 단위로 1개 또는 다수개의 Logical Record로 구성된다.
- File, Data Base

[비트 < 바이트 < 워드 < 필드 < 레코드 < 파일 < 데이터베이스]

주요포인트 꼭 알고 가야해!



❖ 순차 구조(Sequential List)

- 배열(Array)
 - 데이터 형이 동일한 연속적인 Memory의 집합
- Stack (⇒ LIFO)
 - 리스트내의 데이터 삽입, 삭제가 한쪽 끝에서 이루어지는 데이터 구조
 - 구성 : Push(삽입), Pop(삭제), Top, Bottom
 - Stack의 이용 : 부프로그램 호출, 인터럽트 처리, 순환 프로그램, 0주소 명령, 컴파일러
 - 후위 운영(Postorder), 역폴리쉬(후위 표기법) 등에 이용
- Queue(⇒ FIFO)
 - 노드의 삽입 작업은 선형리스트의 한쪽 끝에서 제거 작업은 다른 쪽 끝에서 수행되는 자료구조
 - (1) 구성 : 삭제(front(head)), 삽입(rear(tail))
 - (2) Queue의 이용
 - 운영체제의 작업 스케줄링
 - FIFO(First In First Out) 방식의 작업 스케줄링(Job Scheduling)
 - 일괄처리, 스펙 운영에 이용
- Deque(Double Ended Queue)
 - 삽입과 삭제가 리스트의 양쪽 끝에서 발생할 수 있는 형태이다.

❖ Linked list

- 모든 데이터가 다음 데이터의 위치를 지정하기 위한 포인터를 갖고 있는 구조
- Single linked list
 - Single circular linked list
 - Double linked list.
 - Double Circular linked list

주요포인트 꼭 알고 가야해!

[순환 프로그램(Recursive Program, 재귀적 호출, 되부름 호출)]

자기 자신을 부프로그램 형태로 호출하는 프로그램
 장점 : 프로그램이 간결.
 단점 : 수행속도가 느리다
 (찾은 함수 호출)



[희소행렬(Sparse Matrix)]

원소의 대부분이 0인 행렬로 연속 배열을 이용하는 것보다 Linked List를 이용해 저장하면 기억공간이 절약된다.

3. 비선형구조-Tree

❖ Tree의 정의

- 다음 조건을 만족하는 하나 이상의 node로 구성된 유한집합(finite set)
 - 근이라 부르는 특별히 지정된 1개의 노드가 있다.
 - 근(root) 노드를 제외한 나머지 노드들은 서로 분리된 n개의 부분 집합으로 나누어지며 이들도 또한 Tree이다.
 - Cycle을 배제한 그래프의 특수한 형태

❖ Tree구조의 기본용어

- 근노드(Root node) : 트리의 제일 상위의 레벨에 위치하는 노드.
- 서브 트리(Subtree) : 근노드를 제거했을 때 생기는 트리.
- 차수(Degree) : 어떤 특정 노드의 서브 트리의 수 or 그 노드의 가짓수.
- 단노드(Leaf or Terminal node) : 차수가 0인 노드
- 간노드(Nonterminal node) : 차수가 0이 아닌 노드
- 자노드(Children or son node) : 특정 노드의 서브 트리의 근노드들
- 부노드(Parent or Father node) : 특정 노드가 연결된 바로 상위의 노드
- 제노드(Sibling or Brother node) : 같은 부모를 갖는 자노드들
- 조상(Ancestors) : 근노드로부터 특정 노드에 이르기까지의 경로(path)상에 존재하는 모든 노드들
- 트리 전체의 차수(degree) : 트리 각 노드의 차수 중에서 가장 높은 차수
- 계층(Level) : 근노드를 레벨 1로 하고 최하위 노드까지의 수
- 높이(Height) 및 깊이(Depth) : 트리의 어떤 노드의 최대 레벨.

주요포인트 꼭 알고 가야해!

❖ Binary tree(이진 트리)

- 각 노드의 차수가 2 이하의 트리. 즉 공집합이거나 Root와 Left Subtree or Right Subtree라 불리는 2개의 분리된 이진 트리로 구성된 노드의 유한집합.

❖ Tree와 Binary tree의 차이점

- 각 노드의 차수가 2 이하인 트리
- 각 노드의 위치가 중요한 순서 트리
- 노드의 수가 0인 공백이진 트리도 있을 수 있다.

❖ 특징

- 이진 트리의 레벨 i에서의 최대 노드 수 = 2^{i-1}
- 깊이가 n인 트리의 전체 노드의 최대수 = $2^n - 1$

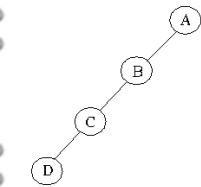
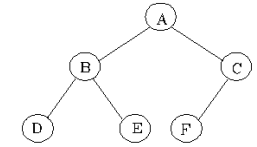
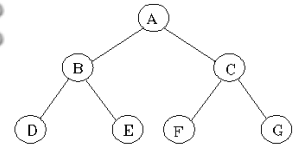
예문) 깊이가 6인 이진트리의 최대 노드 수는 ?

예문) 6개의 단노드가 있는 이진트리에서 차수가 2인 노드의 수는?

❖ 종류

- 정이진 트리(full binary tree)
 - level이 i일 때 그 level의 node 개수가 정확히 2인 tree (depth가 k일 때 2인 tree)
- 전이진 트리(complete binary tree)
 - 깊이가 k이고 전체 노드의 수가 n일 때 1부터 n까지의 노드들이 순서대로 대응되는 트리.
- 사향 트리(skewed tree)
 - 한쪽으로만 치우친 트리.

주요포인트 꼭 알고 가야해!



5. 트리운행법, 수식표기법

❖ 트리의 운행법

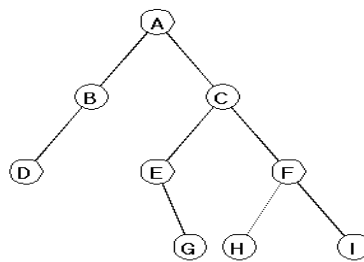
- 트리의 각 노드를 중복되지 않게 한번씩 방문하여 트리의 정보를 검색하는 것.
- Preorder 운행법 (Root → Left → Right)
- Inorder 운행법 (Left → Root → Right)
- Postorder 운행법 (Left → Right → Root)

- 운행법 실습

Preorder : A-B-D-C-E-G-F-H-I

Inorder : D-B-A-E-G-C-H-F-I

Postorder : D-B-G-E-H-I-F-C-A



❖ 수식 표기법

- 산술식을 기억공간에 표현하기 위하여 2진 트리를 사용한다. 연산자의 위치에 따라 수식 표기를 infix, postfix, prefix로 나타낸다. 표기를 했을 때 연산자의 위치는 변경될 뿐 항은 변경되지 않는다.

▪ Infix (중위표기법)

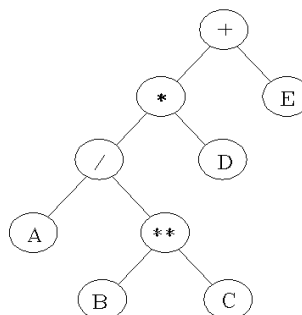
- A / B * C * D + E

▪ Prefix (전위표기법)

- + * / A ** B C D E

▪ Postfix (후위표기법, 역폴리쉬)

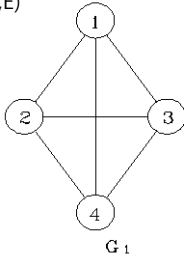
- A B C ** / D * E +



주요포인트 꼭 알고 가야해!

❖ 그래프의 정의

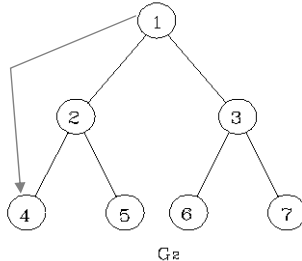
- 공집합이 아닌 정점(Vertex)들의 유한집합 V 와 두 정점들의 쌍인 간선(edge)의 집합인 E 로 구성되는 특수한 자료구조.
- $V(G)$ - 그래프 G 의 정점들의 집합.
- $E(G)$ - 그래프 G 의 간선들의 집합.
- $G=(V,E)$



$$V(G_1) = \{1, 2, 3, 4\}$$

$$V(G_2) = \{1, 2, 3, 4, 5, 6, 7\}$$

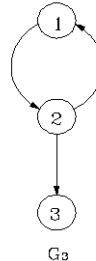
$$V(G_3) = \{1, 2, 3\}$$



$$E(G_1) = \{(1,2), (1,3), (1,4), (2,3), (2,4), (3,4)\}$$

$$E(G_2) = \{(1,2), (1,3), (1,4), (2,5), (3,6), (3,7)\}$$

$$E(G_3) = \{<1,2>, <2,1>, <2,3>\}$$



❖ 관련용어

- ① Undirected graph(무방향 그래프) : 간선을 나타내는 정점들의 쌍의 순서가 없음
- ② Directed graph(방향 그래프) : 간선을 나타내는 정점들의 쌍의 순서가 있는 그래프.
- ③ Complete graph(완전 그래프) : 모든 정점이 1:1로 연결된 그래프, 정점수가 n 일 때 $n(n-1)/2$ 개의 간선을 가진 무방향 그래프
- ④ Cycle : 처음과 끝의 정점이 같은 단순 경로(simple path)
- ⑤ Degree : 정점에 부속하는 간선들의 수

주요포인트 꼭 알고 가야해!

주요포인트 꼭 알고 가야해!



[07강] 정렬과 탐색

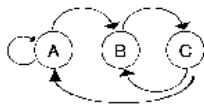
1. 그래프의 표현과 응용
2. 정렬(Sort)
3. 내부정렬의 종류1
4. 내부정렬의 종류2
5. 외부정렬
6. 검색
7. 검색의 종류

합격
비법대한민국 대표 IT강사
조 대 호

여러분의 인생드라마는
지금 잘 진행되시나요?
그 드라마의 주인공은
바로 당신입니다.
이 세상 모든사람들은
당신을 위한 조연이랍니다.
당신의 드라마가
성공한 그날!
당신의 감사메시지 속에서
저의 이름이 불려지기를
기원합니다.

❖인접 행렬(adjacency matrix) 표현법

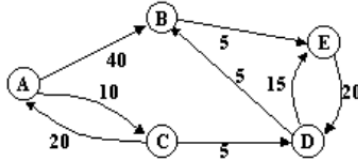
- Row에서는 각 정점의 진출차수를 알 수 있고 Column에서는 진입차수를 알 수 있다.



	A	B	C
A	1	1	0
B	1	0	1
C	0	1	0

❖최단경로 (Shortest Path)

- 가중치가 주어진 방향 그래프 G에서 임의의 정점 v_i 에서 시작하여 정점 v_j 에도 달하는 여러 경로 중에서 가중치의 합이 최소인 경로



정점 A에서 각 정점까지의 최단 경로

A ~ B : A, C, D, B --> 20

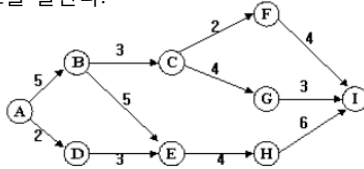
A ~ C : A, C --> 10

A ~ D : A, C, D --> 15

A ~ E : A, C, D, B, E --> 25

❖임계경로 (Critical Path)

- 방향 그래프에서 최장의 경로로써 보통 작업 공정을 나타내는 그래프에 이용
- 작업 공정 그래프에서 각 작업들은 병행 수행되므로 시작 정점에서 완료 정점까지 가장 긴 경로가 다른 작업에 직접적인 영향을 미친다. 따라서 작업을 완료하는 최단 시간은 가장 긴 경로를 말한다.



임계경로 : A, B, E, H, I

(경로길이 : 20)

2. 정렬(Sort)

❖정렬의 개요

- 각 record의 특정 field를 key로 정하고 key값의 순서대로 ascending or descending 순으로 재배열하는 것.

❖정렬의 종류

▪Internal sort (내부정렬)

정렬하고자 하는 소량의 자료를 주기억장치 내에 기억시켜 놓고 재배열하는 방법.

- 삽입법 : insertion sort, shell sort
- 교환법 : bubble sort, quick sort, selection sort
- 선택법 : heap sort
- 분배법 : radix sort
- 병합법 : two way merge sort

▪External sort (외부정렬)

외부기억장소를 이용하여 대량의 자료를 재배열하는 방식으로 기본적으로 병합법(Merge)을 이용한다.

- K-way merge sort
- Balanced sort
- Cascade sort
- Polyphase sort
- Oscillating sort

❖삽입정렬(Insertion sort)

- 기존의 정렬된 순서로 나열된 레코드에 새로운 레코드 R을 삽입하는 방법.

- record의 수가 적을 때 유리
- sorting에 걸리는 시간 : $O(n^2) \Rightarrow$ 평균, 최악

주요포인트 꼭 알고 가야해!

[Big-O 표기법]

알고리즘의 수행시간을 표현하는 방법

특정 문자의 수행횟수를 측정함으로써 평가

$O(1) < O(\log n) < O(n)$
 $< O(n \log n) < O(n^2)$

❖Bubble Sort (Interchange Sort)

- 인접한 2개의 record를 비교해서 크기에 따라 record의 위치를 교환하는 방법으로 뒤에서부터 앞으로 정렬
- ① flag bit를 사용하면 수행 시간을 줄일 수 있다.
- ② 소량의 data에 유리
- ③ program화하기 쉽다.
- ④ sorting time : 평균 $O(n^2)$, 최악 $O(n^2)$

❖Selection sort

- 처음의 key를 선택해 나머지 key와 계속 sort 한 후 두 번째 key와 나머지 key들을 sort 하는 과정을 (n-1) 번째에서는 K_{n-1} 과 K_n 을 비교하여 sort 하는 방법
- ① 짧은 리스트나 자주 정렬되기를 원하지 않는 리스트에 이용
- ② Sorting time : 평균 $O(n^2)$, 최악 $O(n^2)$

❖Quick Sort

- 제어기를 중심으로 작은 값과 큰 값으로 분리하여 한 개의 file로 재배열하는 방법
- ① 평균 수행 시간이 가장 빠르다.
- ② 보조 기억 공간으로 stack을 사용 recursion 이용
- ③ sorting time : 평균 $O(n \log n)$, 최악 $O(n^2)$

❖Two way merge sort

- 이미 순서 배열된 2개의 file을 병합하여 하나의 file로 재배열하는 방법
- ① Sorting time : 평균 $O(n \log n)$, 최악 $O(n \log n)$
- ② 보조 기억 장치에 유리한 방식
- (참조) external two-way merge sort tape 장치 3개 필요

주요포인트 꼭 알고 가야해!

❖Heap sort

- heap이라는 자료 구조를 이용하여 sort하는 방식
- ① heap 구조 : 전이진 트리로서 부 node의 key 값이 자 노드들의 key 값보다 작지 않은 특성을 가진 것
- ② heap sort의 단계
 - ㉠ 주어진 item을 가지고 tree를 구성한다.
 - ㉡ tree를 heap을 구현한다.
 - ㉢ heap의 근노드를 제거하여 출력하고 나머지 tree를 heap이 되도록 조정하는 과정을 반복(모두 제거될 때까지)
- ③ Sorting time 평균 $O(n \log n)$, 최악 $O(n \log n)$

❖Shell sort

- ① insertion sort 방식을 확장한 개념으로 일정한 매개변수에 따라 여러 개의 부 파일로 간주하여 삽입 소트로 배열한다.
- ② 입력 file이 부분적으로 sort 되어 있을 때 좋다.
- ③ 과정
 - ㉠ 매개 변수는 sub-file의 개수를 지정
 - ㉡ 주어진 입력 File을 일정한 매개 변수 값에 따라서 여러 개의 부 file로 간주하며 각 부 File을 insertion sort로 배열
 - ㉢ 매개 변수가 1이 된 후 sort가 끝난다.
- ④ Sorting time : 평균 $O(n^2)$, 최악 $O(n^2)$

❖다중 key에 의한 Sort (= Radix Sort, Bucket Sort, Bin sort)

- Stack이나 Queue를 사용하며, MSD정렬과 LSD정렬이 있다.

주요포인트 꼭 알고 가야해!

MSD(Most Significant Digit first) 정렬

LSD(Least Significant Digit-first) 정렬

❖ External sort(외부정렬)

- K-Way Merge sort
 - 디스크 장치를 이용하여 sort하는 방식
- Balanced sort
 - 4개의 tape를 가지고 two-way merge sort 하는 방식
- Cascade sort
 - n 개(최소 3개 이상)의 tape를 가지고 sort 하는 방식으로 한 개의 tape에 결과가 수록되면 정렬이 끝난다.
- Polyphase sort(다상 정렬)
 - ① 모든 record가 fibonacci 수열을 이루면서 수록되어 있다.
 - ② n-2 개 tape를 동시에 사용
 - ③ 최저 2-way merge를 사용 (최소한 3개의 tape 필요)
- Oscillating sort
 - tape의 역읽기 성질을 이용

❖ Sort 방법을 결정하는데 있어서 고려해야 할 사항

- ① Computer system의 특성
- ② 자료의 양
- ③ 자료의 배열 상태
- ④ 작업 공간의 크기
- ⑤ record의 이동 횟수
- ⑥ key 비교 회수

주요포인트 꼭 알고 가야해!

6. 검색(Search)

❖ Search 정의

- 일정한 기억 공간 내에 저장된 자료 중에서 주어진 조건에 맞는 특정의 자료를 찾는 작업

❖ Searching 방법

- ① 검출 대상 key와 record key를 모두 비교
 - 선형 검색
 - 제어 검색 - Binary search, Fibonacci search, Interpolation search
 - Block 검색
 - Tree 검색
- ② 검색 대상 key의 계수적 성질을 이용하여 비교 없이 계산에 의하여 검색 - Hashing

❖ 선형검색

- 첫 번째 record부터 차례로 비교하면서 원하는 key를 찾는 방법

 - ① 시간이 많이 소모
 - ② 자료가 정렬되어 있지 않아도 사용이 가능하다.
 - ③ 검색 시간 평균 : $O(n)$, 최악 : $O(n)$
 - ④ 평균 비교 횟수 $(n+1)/2$

❖ Binary Search(이진 검색)

- 반드시 Sort 되어 있는 File에서 전체의 파일을 반으로 나누어 가면서 검색

 - ① 레코드가 많을수록 효과가 좋다.
 - ② 순서 배열에 시간이 많이 걸린다.
 - ③ 검색 시간 평균 : $O(\log n)$, 최악 : $O(\log n) + 1 \rightarrow O(\log n)$
 - ④ 한번 비교하는데 시간이 많이 걸린다. (중간 key 값 계산)

주요포인트 꼭 알고 가야해!

예문> 아래 자료에서 65를 찾기 위하여 2진 검색할 경우 비교해야 할 횟수는?

3, 18, 47, 54, 65, 83, 94, 97

❖Fibonacci Search

▪ Fibonacci 수열에 따라 다음에 비교할 대상을 선정하여 검출을 진행하는 방법

① Fibonacci 수열

$F_0=0, F_1=1, \dots$: fibonacci 수열의 일반식 $\rightarrow F_i=F_{i-1} + F_{i-2} (i \geq 2)$

$\rightarrow 0, 1, 1, 2, 3, 5, 8, 13, 21, 34, \dots$

② 평균 검출은 이진 검색보다 빠르나 최악의 경우는 느리다.

③ 검색 시간 평균 : $O(\log n)$, 최악 : $O(\log n)$

❖Interpolation search(보간 검색)

▪ 처음에 비교할 대상을 선정할 때 처음부터 주어진 key와 일치하는 것이 있음직한 부분의 key를 선택하여 검색.

▪ ex) 사전 찾기

▪ 계산식이 복잡하여 잘 사용되지 않는다.

❖Block search

▪ block 단위로 record를 나누어 저장하고 검색하고자 하는 자료가 어떤 block에 있는지 찾은 후 해당 block 내에서 선형 검색을 하는 방법

❖Binary tree search

▪ 파일을 이진 검색 트리로 구성하여 이진 검색과 같은 시간에 검색이 이루어지도록 한다.

주요포인트 꼭 알고 가야해!

주요포인트 꼭 알고 가야해!

(GO)

[08강] 파일 구조

1. 순차파일과 직접파일
2. 색인순차파일
3. 다중키 파일
4. 인덱스구조
5. 해싱
6. 데이터웨어하우스와 OLAP
7. 전산영어

합격
비법



대한민국 대표 IT강사
조 대 호

여러분의 인생드라마는
지금 잘 진행되시나요?
그 드라마의 주인공은
바로 당신입니다.
이 세상 모든사람들은
당신을 위한 조연입니다.
당신의 드라마가
성공한 그날!
당신의 감사메시지 속에서
저의 이름이 불려지기를
기원합니다.

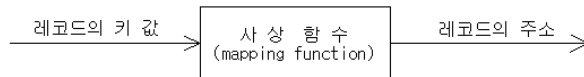
❖ Sequential file(순차 file : SAM)

- 레코드들의 논리적인 순서와 기억 매체에 수록된 물리적인 순서가 일치되도록 편성하는 방법으로 생성되는 순서에 따라 레코드를 순차적으로 저장하므로 저장 매체의 효율이 가장 높다.

- 장점
 - 기억장소의 낭비가 없다.
 - 순서대로 저장되어 취급이 용이하다.
 - 여러 매체에 쉽게 수용
- 단점
 - 순서적 처리만 가능하다.
 - 삽입삭제 시 시간낭비가 심하다.
- 일괄 처리의 상업적 업무에 이용
- 활동율이 높은 경우 사용.

❖ Direct file(직접 편성 파일 : DAM)

- 기억 공간의 장소에 순서 없이 무작위로 직접 수록하는 방법



- 장점
 - 특정 레코드에 대한 직접 접근이 가능하므로 대화식 처리에 이용된다.
 - 어떤 레코드도 평균 접근 시간으로 검색이 가능하므로 속도가 빠르다.
- 단점
 - 계산으로 번지를 결정하므로 계산방법에 따라 기억공간의 낭비가 많다.
 - 충돌 현상이 발생했을 때 처리 방법을 결정해야 한다.

주요포인트 꼭 알고 가야해!

2. 색인순차파일(ISAM)

❖ Indexed sequential file(ISAM)

- 레코드가 키(key) 순서대로 기억되며 별도로 키에 대응되는 주소로써 색인(index)을 만들어 파일의 처리를 직접처리와 순차처리가 가능한 편성법이다.

❖ ISAM의 구성요소

- 기본 데이터 영역(Prime Data area)
 - 실제의 데이터 레코드가 기록되며 반드시 키 값의 순서에 의해 분류되어 있어야 한다.
- 색인 영역(Index area)
 - 기본 데이터 영역에 대한 색인을 구성하는 부분으로 탐색시간을 줄이기 위해 계층적 구조를 사용한다.
 - 마스터 인덱스
 - 실린더 인덱스
 - 트랙 인덱스
- 오버플로우 영역(Overflow area)
 - 정적 인덱스 편성인 경우 추가로 삽입되는 레코드가 기본 데이터 구역에 기록되지 못하기 때문에, 이 넘친 데이터를 기록하는 부분이다.
 - 실린더(cylinder)오버플로우 영역
 - 독립(independent)된 오버플로우 영역

❖ ISAM의 장단점

- 장점
 - 순서처리와 직접처리가 가능하다.
 - 파일내의 레코드를 추가, 삭제할 경우 파일전체를 복사할 필요가 없다.
- 단점
 - 주기적으로 재구성해야 한다.
 - 기억 공간 낭비가 발생한다.

주요포인트 꼭 알고 가야해!

❖다중 키 파일의 정의

- 하나의 데이터 레코드 집합에 여러 개의 상이한 접근 방법을 지원하는 구조로써 이 다중 키 파일의 기본 개념은 데이터를 중복시키지 않고 다중 경로를 제공하는 것이다.

주요포인트 꼭 알고 가야해!

❖다중 키 파일의 종류

(1) Inverted file(역파일)

배열 순서를 반대로 한 파일 구조로서 색인과 데이터 파일 사이의 연결을 위하여 데이터 레코드 및 키 값 전체에 대한 포인터를 가지고 있으며, 정보 검색 시 키워드(keyword)로 해당 데이터 레코드를 찾을 수 있도록 상호 참조 파일을 편성하는 방법이다.

※ 특징

- ① 데이터 레코드 및 키 값 전체에 대한 포인터를 가진다.
- ② Index의 길이가 가변적이다.
- ③ 질의 수행 능력이 우수하다

(2) Multilist file(다중 리스트 파일)

특정 필드의 키 값이 같은 데이터 레코드들로 구성된 연결 리스트(linked list)로 이루어진 파일 구조로서 보조 키 값에 관련된 레코드들의 검색을 빠르게 하기 위해서 연결 리스트에 대한 색인을 가진다.

※ 특징

- ① 색인을 구성하는 각 엔트리들이 고정 길이를 가지므로 관리가 용이하다
- ② 데이터의 구성에 인덱스가 영향을 준다.
- ③ 같은 내용을 갖는 데이터 처리에 효과적이다.

4. 인덱스 구조

❖인덱스 구성 방법

① 정적 인덱스(ISAM)

데이터 파일에 레코드가 삽입되거나 삭제될 때 인덱스의 내용은 변하고 구조는 변경되지 않는다.

② 동적 인덱스(VSAM)

인덱스나 데이터 파일을 블록으로 구성하고 각 블록에는 추가로 삽입될 레코드를 감안하여 빈 공간을 미리 예비해 두는 방법으로 오버플로우 영역이 필요 없다.

❖VSAM의 장점

- 기본데이터구역과 오버플로우구역을 구분하지 않으며 ISAM은 논리적삭제에 반해 VSAM은 물리적삭제를 하고 ISAM은 고정길이레코드만 수용하지만 VSAM은 가변길이 레코드를 수용할 수 있다

❖B-트리

- 인덱스를 조직하는 방법으로 가장 많이 사용되는 구조이며, 균형된 탐색 트리로서 효율적인 균형 알고리즘을 제공한다.

❖B+-트리

- B+트리는 B-트리의 순차접근의 취약점을 보완한 트리로서 리프노드를 찾기 위하여 경로를 제공하는 인덱스 부분(index set)과 트리의 내부노드를 포함하여 모든 노드를 순차적으로 리프노드에 열거해 놓은 순차부분(sequence set)으로 구성한다.
- B-트리와의 차이점
 - 인덱스 부분은 경로로만 사용되고, 인덱스 부분의 모든 키는 순차세트에 나타나게 되며, 순차세트는 순차적으로 연결되어 있으며, 리프노드와 내부노드는 다른 구조를 갖는다.

주요포인트 꼭 알고 가야해!

❖ Hashing의 정의

- 연산 함수에 의하여 주어진 data 들과 table address들 간의 관계를 갖는 table을 구성한 후 그 function을 이용하여 search 하는 방법

❖ 특징

- ① 각 record의 keyword를 일일이 비교해서 찾을 필요가 없다.
- ② space는 많이 차지하지만 속도는 더 빠르다.
- ③ 주소결정을 위한 키 변환 과정이 필요하다
- ④ 기억공간의 효율이 저하된다.

❖ Hashing 관련용어

- collision (충돌 = crash) : 레코드를 삽입할 때 2개의 상이한 레코드가 똑같은 버킷으로 해싱 되는 것
- Synonym : 해싱 함수의 값을 구한 결과 키 K1, K2가 같은 값을 가질 때, 이들 키 K1, K2의 집합
- Bucket : 하나의 주소를 갖는 파일의 한 구역을 의미한다
- Slot : 한 개의 레코드를 저장할 수 있는 공간으로 n개의 슬롯이 모여 하나의 버킷을 형성한다.
- Overflow : 계산된 home address의 bucket 내에 저장할 기억공간이 없는 상태, 빈 slot이 없는 상태

❖ hashing function의 종류 (home address를 구하는 방법)

- ① Division method(제산)
- ② Mid-square method(제곱)
- ③ Folding(접지법)
- ④ Radix conversion method(진법 변환)
- ⑤ Algebraic coding method(다항식의 제산)
- ⑥ Digit analysis method(계수 분석)
- ⑦ Exclusive OR method
- ⑧ Random method

주요포인트 꼭 알고 가야해!

Hashing 함수가 갖추어야 할 조건

- ① 함수 계산이 쉬워야 한다.
- ② collision을 최소화하여야 한다.

❖ 데이터웨어하우스의 정의

- 의사결정지원을 위한 주제 지향의 통합적이고 영속적이면서 시간에 따라 변하는 데이터의 집합.
- 여러 소스의 데이터를 수집해 하나의 통일된 스키마를 이용하여 단일 사이트에 저장한 정보 저장소

❖ 데이터웨어하우스의 특징

- 복잡한 분석, 지식 발견, 의사결정지원을 위한 데이터의 접근을 제공
- 의사결정지원을 위한 데이터는 장기간 보관되며 판독 전용(Read Only)로 사용된다

❖ OLAP(On Line Analytical Processing)

- 대규모의 다차원 데이터를 동적으로 온라인에서 분석,통합하고 보고서를 만드는 과정
- OLAP을 위한 데이터는 다차원배열로 저장되어 있는 것으로 취급하고 처리함

❖ OLAP의 종류와 특징

- 데이터웨어하우스에 저장된 데이터는 보통 대규모이므로 필요한 정보를 생성하기 위해서는 데이터를 여러가지 형태로 그룹핑할 것이 요구된다.
- 데이터를 다차원 배열로 표현한 것을 다차원 데이터라고 함
- 종류 : ROLAP, MOLAP

❖ 데이터마이닝

- 대량의 데이터로부터 관련된 정보를 발견하는 과정
- 체계적이고 자동적으로 데이터로부터 통계적 규칙이나 패턴을 찾아감

❖ XML 데이터베이스

- 데이터교환을 위한 표준안으로 현재는 다양하게 확대해서 사용하고 있음

주요포인트 꼭 알고 가야해!

주요포인트 꼭 알고 가야해!



[전산영어문제 해결방안]

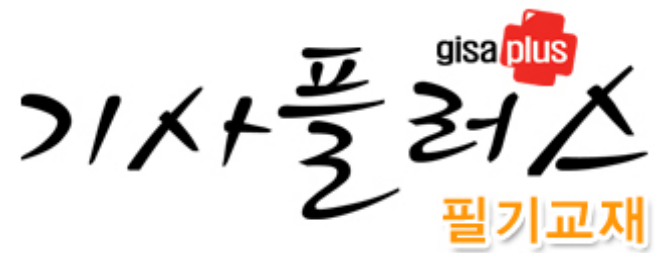
1. 데이터베이스 전체문제중에서 가장 쉽다.
2. 문장중에 키워드를 찾아라
3. 이니셜이 뭔지를 유추하라
4. 자신감을 가지고 또박또박 읽어나가자~!



대한민국 대표 IT강사
조 대 호

여러분의 인생드라마는
지금 잘 진행되시나요?
그 드라마의 주인공은
바로 당신입니다.
이 세상 모든사람들은
당신을 위한 조연이랍니다.
당신의 드라마가
성공한 그날!
당신의 감사메시지 속에서
저의 이름이 불려지기를
기원합니다.

정보처리기사/산업기사 합격은 당연! 고득점 합격이 목표! (GO)



정보처리기사/산업기사 **필기**

전 자 계 산 기 구조



주요포인트 꼭 알고 가야해!

**[01강] 논리회로**

1. 기본논리회로(AND, OR, NOT)
2. NAND, NOR, XOR, NXOR
3. 불대수
4. 간소화(간략화)

대한민국 대표 IT강사
조 대 호

여러분의 인생드라마는
지금 잘 진행되시나요?
그 드라마의 주인공은
바로 당신입니다.
이 세상 모든사람들은
당신을 위한 조연이랍니다.
당신의 드라마가
성공한 그날!
당신의 감사메시지 속에서
저의 이름이 불려지기를
기원합니다.

정보처리기사/산업기사 합격은 당연! 고득점 합격이 목표! (GO)

1. 기본논리회로(AND,OR,NOT)

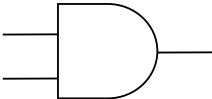
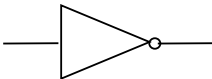
JODAEHO

gisa plus

❖ 기본 논리회로

하드웨어를 구성하는 기본 소자(게이트)의 기능을 수학적으로 표현.
게이트 종류: AND, OR, NOT, NAND, NOR, XOR, XNOR

❖ AND($\cap$, *), OR($\cup$, +), NOT

종류	논리식	회로도	진리표															
AND	$X = A \cdot B$		<table><tr><th>A</th><th>B</th><th>X</th></tr><tr><td>0</td><td>0</td><td>0</td></tr><tr><td>0</td><td>1</td><td>0</td></tr><tr><td>1</td><td>0</td><td>0</td></tr><tr><td>1</td><td>1</td><td>1</td></tr></table>	A	B	X	0	0	0	0	1	0	1	0	0	1	1	1
A	B	X																
0	0	0																
0	1	0																
1	0	0																
1	1	1																
OR	$X = A + B$		<table><tr><th>A</th><th>B</th><th>X</th></tr><tr><td>0</td><td>0</td><td>0</td></tr><tr><td>0</td><td>1</td><td>1</td></tr><tr><td>1</td><td>0</td><td>1</td></tr><tr><td>1</td><td>1</td><td>1</td></tr></table>	A	B	X	0	0	0	0	1	1	1	0	1	1	1	1
A	B	X																
0	0	0																
0	1	1																
1	0	1																
1	1	1																
NOT	$X = A'$		<table><tr><th>A</th><th>X</th></tr><tr><td>0</td><td>1</td></tr><tr><td>1</td><td>0</td></tr></table>	A	X	0	1	1	0									
A	X																	
0	1																	
1	0																	

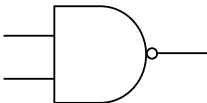
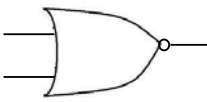
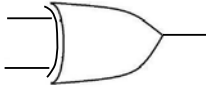
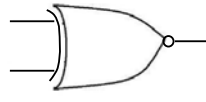
주요포인트 꼭 알고 가야해!

정보처리기사/산업기사 합격은 당연! 고득점 합격이 목표!

**[01강] 논리회로**

Slides 2

❖ NAND, NOR, XOR, XNOR

종류	논리식	회로도	진리표															
NAND	$X = \overline{A \cdot B}$		<table><tr><th>A</th><th>B</th><th>X</th></tr><tr><td>0</td><td>0</td><td>1</td></tr><tr><td>0</td><td>1</td><td>1</td></tr><tr><td>1</td><td>0</td><td>1</td></tr><tr><td>1</td><td>1</td><td>0</td></tr></table>	A	B	X	0	0	1	0	1	1	1	0	1	1	1	0
A	B	X																
0	0	1																
0	1	1																
1	0	1																
1	1	0																
NOR	$X = \overline{A + B}$		<table><tr><th>A</th><th>B</th><th>X</th></tr><tr><td>0</td><td>0</td><td>1</td></tr><tr><td>0</td><td>1</td><td>0</td></tr><tr><td>1</td><td>0</td><td>0</td></tr><tr><td>1</td><td>1</td><td>0</td></tr></table>	A	B	X	0	0	1	0	1	0	1	0	0	1	1	0
A	B	X																
0	0	1																
0	1	0																
1	0	0																
1	1	0																
XOR	$X = A'B + AB'$ $X = A \oplus B$		<table><tr><th>A</th><th>B</th><th>X</th></tr><tr><td>0</td><td>0</td><td>0</td></tr><tr><td>0</td><td>1</td><td>1</td></tr><tr><td>1</td><td>0</td><td>1</td></tr><tr><td>1</td><td>1</td><td>0</td></tr></table>	A	B	X	0	0	0	0	1	1	1	0	1	1	1	0
A	B	X																
0	0	0																
0	1	1																
1	0	1																
1	1	0																
NXOR	$X = A' \cdot B' + A \cdot B$ $X = A \odot B$		<table><tr><th>A</th><th>B</th><th>X</th></tr><tr><td>0</td><td>0</td><td>1</td></tr><tr><td>0</td><td>1</td><td>0</td></tr><tr><td>1</td><td>0</td><td>0</td></tr><tr><td>1</td><td>1</td><td>1</td></tr></table>	A	B	X	0	0	1	0	1	0	1	0	0	1	1	1
A	B	X																
0	0	1																
0	1	0																
1	0	0																
1	1	1																

주요포인트 꼭 알고 가야해!

[Exclusive OR Gate의 용도]

- ① 두 수의 합을 구하는 회로 (가산기의 한 부분)
- ② overflow 판정
- ③ 비교기
- ④ 짝수 Parity 검출

[Exclusive NOR Gate의 용도]

- ① 비교기
- ② 홀수 Parity 검출

3. 불 대수

❖ 불 대수(Boolean Algebra)

2진수를 이용하여 참 또는 거짓의 논리적인 판단을 하는 수학적 방법.
컴퓨터 작동의 논리적 풀이 방법.

▶ 기본연산: AND($\cap$, $\cdot$), OR($\cup$, $+$), NOT(여집합), 0: 공집합, 1: 전체집합

A·B	F	A+B	F	A	NOT	F
0 0	0	0 0	0	0		1
0 1	0	0 1	1	1		0
1 0	0	1 0	1			
1 1	1	1 1	1			

▶ 법칙: 일반, 교환, 결합, 분배, 드모르간

불 대수 기본 법칙			
일반 법칙	$A+A = A$, $A \cdot A = A$, $A+A' = 1$, $A \cdot A' = 0$, $1+A = 1$, $1A = A$, $0A = 0$, $(A')' = A$		
교환 법칙	$A+B = B+A$	$A \cdot B = B \cdot A$	
결합 법칙	$A+(B+C) = (A+B)+C$	$A \cdot (B \cdot C) = (A \cdot B) \cdot C$	
분배 법칙	$A \cdot (B+C) = A \cdot B + A \cdot C$	$A+B \cdot C = (A+B) \cdot (A+C)$	
드모르간 법칙	$\overline{A+B} = \overline{A} \cdot \overline{B}$	$(A+B)' = A' \cdot B'$	$\overline{A \cdot B} = \overline{A} + \overline{B}$
			$A' \cdot B' = (A+B)'$

주요포인트 꼭 알고 가야해!

❖ 불 대수를 이용한 간소화

$$A + A \cdot B = A \cdot (1 + B) = A \cdot 1 = A$$

$$A(A + B) = A \cdot A + A \cdot B = A + A \cdot B = A \cdot (1 + B) = A \cdot 1 = A$$

$$A + A' \cdot B = (A + A')(A + B) = 1 \cdot (A + B) = A + B$$

$$A(A' + B) = A \cdot A' + A \cdot B = 0 + A \cdot B = A \cdot B$$

$$\begin{aligned} X &= A \cdot B + A \cdot B' + A' \cdot B \\ &= A \cdot (B + B') + A' \cdot B \\ &= A \cdot 1 + A' \cdot B \\ &= A + A' \cdot B \\ &= (A + A') \cdot (A + B) \\ &= 1 \cdot (A + B) \\ &= A + B \end{aligned}$$

❖ 도시법(Karnaugh-map)을 이용한 간소화

- ① 출력이 1인 항을 직사각형 형태로 묶되, 2의 거듭제곱의 수로 묶는다..
- ② 같은 묶음의 변수끼리는 AND, 서로 다른 묶음의 결과들끼리는 OR연산을 한다.

[2변수 Karnaugh-map]

A \ B	B	B'	B
A'	1	2	
A	3	4	

[3변수 Karnaugh-map]

A \ BC	B'C'	B'C	BC	BC'
A'	1	2	4	3
A	5	6	8	7

예제1>

A \ BC	00	01	11	10
0	0	0	1	1
1	0	1	1	1

예제2>

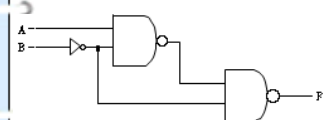
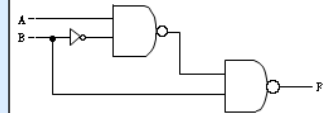
A \ BC	00	01	11	10
0	1	0	0	1
1	1	1	1	1

주요포인트 꼭 알고 가야해!

예문) 불대수 식의 정리 중 옳지 않은 것은?

- ① $A + A \cdot B = A$
- ② $A + A' \cdot B = A + B$
- ③ $A + 0 = A$
- ④ $A \cdot (A' + A \cdot B) = A + B$

[불대수를 이용한 간소화]



제2과목 - 전자계산기구조

주요포인트 꼭 알고 가야해!

GO

[02강] 자료의 표현

1. 조합논리회로
2. 순서논리회로
3. 진법변환
4. 자료표현의 종류

합격
비법대한민국 대표 IT강사
조 대 호

여러분의 인생드라마는
지금 잘 진행되시나요?
그 드라마의 주인공은
바로 당신입니다.
이 세상 모든사람들은
당신을 위한 조연이랍니다.
당신의 드라마가
성공한 그날!
당신의 감사메시지 속에서
저의 이름이 불려지기를
기원합니다.

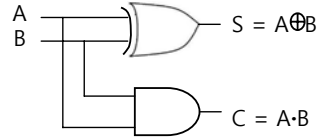
❖ 조합 논리회로

- 기본논리회로 2개 이상 조합된 회로. 입력값에 의해서만 출력값 결정. 기억 기능 없음.
- 종류 : 반가산기, 전가산기, 감산기, 인코더, 디코더, 멀티플렉서, 디멀티플렉서

❖ 반가산기(XOR 1개, AND 1개)

- 2진수로 나타낸 수들을 1bit씩 합하여 그 결과로 1bit의 합(sum)과 1bit의 올림수(carry)를 발생시키는 회로.

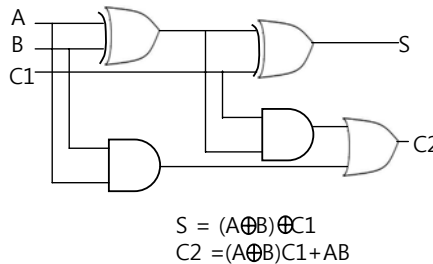
A	B	S	C
0	0	0	0
0	1	1	0
1	0	1	0
1	1	0	1



❖ 전가산기(반가산기 2개, OR 1개)

- 2개의 입력 비트와 앞단에서 발생된 carry까지를 고려하여 3개의 bit를 가산하는 회로

A	B	C1	S	C2
0	0	0	0	0
0	0	1	1	0
0	1	0	1	0
0	1	1	0	1
1	0	0	1	0
1	0	1	0	1
1	1	0	0	1
1	1	1	1	1



주요포인트 꼭 알고 가야해!

[인코더(Encoder)]

암호로 변환하는 장치
2진 신호로 변환
입력 2^n 개 → 출력 n 개

[디코더(Decoder)]

암호 해독 장치
입력 n 개 → 출력 2^n 개

[멀티플렉서(MUX)]

2^n 개의 입력 중
1개 선택하여 출력

[디멀티플렉서(DeMUX)]

1개의 입력을 2^n 개 중
1개 선택하여 출력

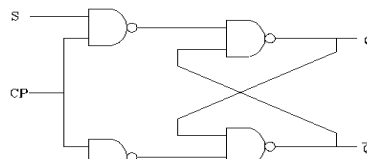
2. 순서논리회로

❖ 순서 논리회로

- 입력과 현재 상태에 따라 출력 결정. 기억기능 있음. 조합회로와 플립플롭으로 구성.
- 플립플롭(1개의 Bit 저장. 상태 변화 신호가 있을 때 까지 현 상태 유지)

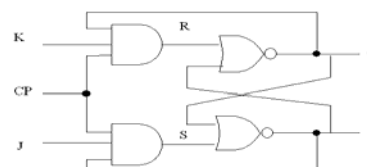
▶ RS 플립플롭

S	R	Q
0	0	변화 없음
0	1	0
1	0	1
1	1	동작 안 함



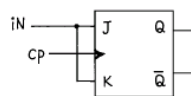
▶ JK 플립플롭(RS 보완, 가장 많이 사용)

J	K	Q
0	0	변화 없음
0	1	0
1	0	1
1	1	반전(보수)



▶ T 플립플롭(입력 한 개. ON,OFF 교차 스위치)

D	Q
0	변화 없음
1	반전(토글)



▶ D 플립플롭(입력 한 개. 입력값 그대로 저장)

D	Q
0	0
1	1

주요포인트 꼭 알고 가야해!

[Race문제]

Flip-Flop의 출력측이 입력측으로 Feedback되기 때문에 CP=1일 때 출력측의 값이 변화하면 입력측의 값이 변화되어 오동작을 발생하게 된다. 이 오동작이 다른 오동작을 유발하게 되는데 이러한 현상을 Race현상이라 한다.

이러한 Race현상을 없애는 방법은 Edge triggered F/F, Master slave F/F를 사용하면 된다.

❖진수 / 변환

- 진수 종류: 2진수(0,1), 8진수(0~7), 16진수(0~9,A~F), 10진수(0~9)
- 진수(법) 변환(정수)
 - 10진수 → 2,8,16진수
각 진수로 나눌 수 없을 때 까지 나눈 후 나머지 값을 역순으로 표기.
 - 2,8,16진수 → 10진수
각 진수의 가중치를 모두 더하여 표기.
 - 2진수 ↔ 8진수, 16진수
2진수 3자리는 8진수 1자리 / 16진수 4자리로 표기.
 - 8진수 ↔ 16진수
8진수 ↔ 16진수는 2진수로 변환 후 각 진수로 표기.

진수(법) 변환(실수)

 $26.125_{(10)} \rightarrow 11010.001_{(2)}$

$$\begin{array}{r} .125 \\ \times 2 \\ \hline 0.250 \\ \times 2 \\ \hline 0.500 \\ \times 2 \\ \hline 1.000 \end{array}$$

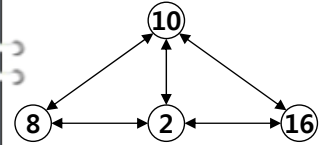
 $26.625_{(10)} \rightarrow 11010.101_{(2)}$

$$\begin{array}{r} .625 \\ \times 2 \\ \hline 1.250 \\ \times 2 \\ \hline 0.500 \\ \times 2 \\ \hline 1.000 \end{array}$$

 $26.125_{(10)} - 8\text{진수로 변경} \rightarrow 32.1_{(8)}$

$$\begin{array}{r} .125 \\ \times 8 \\ \hline 1.000 \end{array}$$

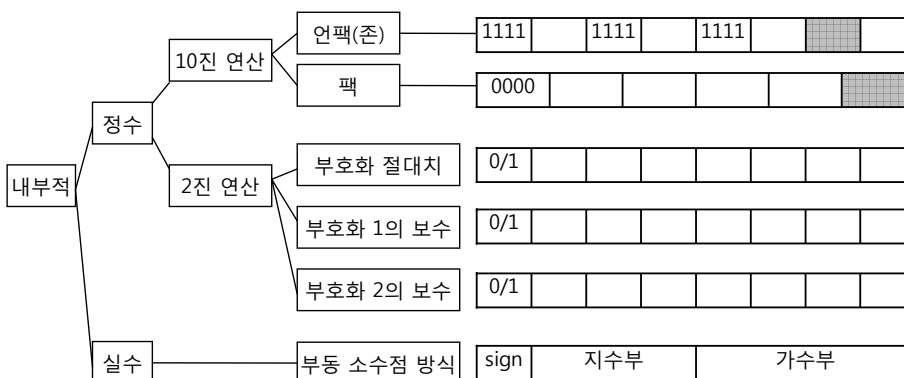
주요포인트 꼭 알고 가야해!



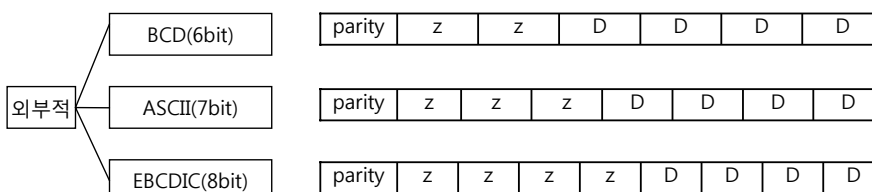
4. 자료표현의 종류

❖자료 표현의 종류

- 숫자표현



- 문자표현



주요포인트 꼭 알고 가야해!

양수(C) : 1100
음수(D) : 1101

주요포인트 꼭 알고 가야해!

(GO)

[03강] 자료의 연산

1. 고정소수점
2. 부동소수점, 문자표현
3. 가중, 비가중 코드
4. 자료의 연산
5. 고정, 부동소수점 연산

합격
비법대한민국 대표 IT강사
조 대 호

여러분의 인생드라마는
지금 잘 진행되시나요?
그 드라마의 주인공은
바로 당신입니다.
이 세상 모든사람들은
당신을 위한 조연이랍니다.
당신의 드라마가
성공한 그날!
당신의 감사메시지 속에서
저의 이름이 불려지기를
기원합니다.

정보처리기사/산업기사

합격은 당연! 고득점 합격이 목표!

(GO)

1. 고정소수점

JODAEHO

gisa plus

❖ 고정소수점의 표현

■ 정수를 2진수 형태로 표현하는 방법으로 양수의 경우 모두 동일하나 음수의 경우 부호와 절대치, 1의 보수, 2의 보수의 경우가 모두 다르게 표현된다.

0	1	2		n-1
S	Data부분			

S는 부호를 의미(0:양수, 1:음수)

(1) 수의 표현 및 표현 범위

음수의 표현 방식	
부호와 절대치	부호 비트와 그 크기를 나타내는 절대값으로 구성하는 방식 가장 왼쪽 비트를 부호로 사용(0 : 양수, 1 : 음수)
부호와 1의 보수	모든 비트에 대해 0은 1로, 1은 0으로 바꿔줌
부호와 2의 보수	1의 보수로 바꿔준 다음 2진수 1을 더해줌

예) +13과 -13의 표현(8비트인 경우)

양수 표현	0 0 0 0 1 1 0 1	표현범위(n bit의 경우)
부호와 절대치	1 0 0 0 1 1 0 1	
부호와 1의 보수	1 1 1 1 0 0 1 0	$-(2^{n-1}-1) \sim 2^{n-1}-1$
부호와 2의 보수	1 1 1 1 0 0 1 1	$-2^{n-1} \sim 2^{n-1}-1$

(2) 2의 보수 사용 이유

- 부호와 절대치와 1의 보수에서는 0의 표현이 +0과 -0이 다르게 표현
- 부호와 2의 보수에서는 0의 표현이 유일
- 부호와 2의 보수에서는 음수에서 한 개를 더 표현하는 것이 가능하다.
- 연산 속도가 더 빠르다.

주요포인트 꼭 알고 가야해!

정보처리기사/산업기사

합격은 당연! 고득점 합격이 목표!

(GO)

[03강] 자료의 연산

Slides 12

❖ 실수의 표현(부동소수점, Floating-point)

- 2진 실수 데이터 표현과 연산에 사용된다. 지수부(Exponent)와 가수부(Mantissa)로 구성되며 고정소수점보다 복잡하고 실행시간이 많이 걸리나 아주 큰 수나 작은 수의 표현이 가능하다.
- 소수점은 지수부와 가수부 사이(7번 비트와 8번 비트 사이)에 있는 것으로 묵시적으로 간주하고 지수부와 가수부로 분리시키는 정규화(Normalization) 과정이 필요하며 지수에 상수 64인 바이어스(Bias)를 더해서 표현한다.



❖ 문자자료 표현

- BCD(Binary Coded Decimal) code - 이진화 십진수로 나타낸 숫자 코드
 - 하나의 문자를 6개의 데이터 비트와 1개의 패리티 비트로 구성하여 64개의 문자를 나타낼 수 있도록 한 문자 코드로 6비트 중 앞의 2개는 존(zone) 비트, 뒤의 4비트는 디지트(digit) 비트이다.
 - 존 비트는 숫자는 00, A-I는 11, J-R이 10, S-Z가 01이다.
- ASCII(American Standard Code for Information Interchange) code
 - 미국 표준 협회가 제정한 자료 처리 및 통신 시스템 상호 간의 정보 교환용 표준 코드로 7비트로 구성되어 총 128가지의 코드가 있다. 이들은 제어 부호 33자, 그래픽 부호 33자, 숫자 10자, 알파벳 대소문자 52자로 되어 있고 대부분의 컴퓨터에서 내부적으로 문자를 표시하는 데 사용되고 있으며 데이터 통신에서도 많이 이용되고 있다
- EBCDIC(Extended BCD Interchange Code) code
 - 주로 컴퓨터 내부에 문자를 이진수로 표현하기 위해 사용하는 코드의 일종으로 8비트 코드로 256개의 다른 문자를 나타낼 수 있으며 주로 IBM 대형 컴퓨터에서 기본 코드로 사용되고 4개의 zone bit와 4개의 digit bit로 구성

주요포인트 꼭 알고 가야해!

[부동소수점 표현의 특징]

- 고정 소수점 표현보다 표현의 정밀도를 높일 수 있다.
- 아주 작은 수와 아주 큰 수의 표현에 적합하다.
- 연산 속도가 느리다
- 수 표현에 필요한 자리에 있어서 효율적이다.
- 과학이나 공학 또는 수학적인 응용에 주로 사용되는 수 표현이다.

주요포인트 꼭 알고 가야해!

BCD(8421) 코드	대표적 가중치 코드. 145 -> 0001 0100 0101																								
Excess-3(3초과) 코드	BCD + 3, 자기보수 코드, 비가중치 코드. 10 -> 1010 + 11 -> 1101																								
Gray 코드	XOR 연산, 아날로그 -> 디지털 변환에 주로 사용. 하드웨어 에러를 최소화하는데 사용 <table><tr><th colspan="4">2진수 -> gray</th><th colspan="4">gray -> 2진수</th></tr><tr><td>1</td><td>0</td><td>1</td><td>0</td><td>1</td><td>1</td><td>1</td><td>0</td></tr><tr><td>1</td><td>1</td><td>1</td><td>1</td><td>1</td><td>0</td><td>1</td><td>1</td></tr></table>	2진수 -> gray				gray -> 2진수				1	0	1	0	1	1	1	0	1	1	1	1	1	0	1	1
2진수 -> gray				gray -> 2진수																					
1	0	1	0	1	1	1	0																		
1	1	1	1	1	0	1	1																		
패리티 검사 코드	오류 검사위해 1bit(패리티 체크비트) 추가. 짝수(Even) 패리티 : 10101000 => 101010001, 10101010 => 101010100 홀수(Odd) 패리티 : 10101000 => 101010000, 10101010 => 101010101																								
해밍코드	오류 검출 교정. 1,2,4 ... 2 ⁿ 자리(좌에서 우로)는 패리티비트. 11001010																								

[가중치 코드(Weighted code)]

- 8421 code
- BCD 코드
- 2421 code
- 자보수 code
- 5421 code
- 51111 code
- 7421 code
- 74-2-1 code

[비가중치 코드(Non-Weighted code)]

- Access-3 code(3초과코드)
- Gray code

❖ 연산자의 기능

- 함수 연산 기능
 - Load : 주기억장치의 데이터를 CPU의 레지스터로 읽어 들이는 명령
 - Store : 레지스터의 데이터를 주기억장치로 기억시키는 명령
- 제어 기능
- 입출력 기능

❖ 수치 연산

- 단항: ROTATE, SHIFT, MOVE, NOT / 이항: AND, OR, 사칙연산
- 음수 연산(보수)
 - 1의 보수 : 0 → 1, 1 → 0 으로 변환.
 - 2의 보수 : 1의 보수로 변환 후 1을 더함.
- 곱하기, 나누기 연산(shift)
 - left shift : 좌측 이동. 00000101 → 좌로 2비트 이동 ⇒ 20
 - right shift : 우측 이동. 00000101 → 우로 2비트 이동 ⇒ 1

❖ 논리 연산

- AND (=Masking Operation)
 - AND는 비수치적인 자료 중 필요 없는 부분을 삭제하려 할 때 사용되는 연산.
- OR (=Selective Set)
 - OR는 AND와는 반대로 비수치적인 자료 중에서 필요한 부분에 내용을 삽입하기 위해서 사용하는 연산이다.
- MOVE : 레지스터의 값을 다른 레지스터로 이동.
- ROTATE : shift 연산 시 밀려나간 비트가 반대쪽으로 들어감.

주요포인트 꼭 알고 가야해!

[예문]

A = 1100, B = 1010 일 경우
다음의 연산 결과는?

① AND 연산

② OR 연산

③ XOR 연산

5. 고정, 부동소수점의 연산

❖ 고정소수점 연산

- (1) 덧셈
부호화 된 1의 보수 및 2의 보수 형태로 표시된 두 수의 덧셈은 부호 비트를 포함하여 더한다. 이때 부호 비트로부터 나오는 carry를 1의 보수 방법에서는 그 carry를 최우측 비트에 더해주며(End Around Carry) 2의 보수 방법에서는 그 carry를 무시한다.
- (2) 뺄셈
감수의 보수를 취해 더하는 방법으로 계산한다.
- (3) 곱셈과 나눗셈
곱셈과 나눗셈은 별도의 연산기를 사용하지 않고 산술적인 Shift에 의해 수행한다. 오른쪽 Shift는 나눗셈을 수행하는 의미이고 왼쪽 Shift는 곱셈을 수행하는 의미이다. 왼쪽 Shift의 경우에는 부호 비트가 반전되면서 Shift되면 오버플로우가 발생하므로 주의해야 하며 오른쪽 Shift의 경우에는 밀려나가는 비트가 1이면 잘림이 발생한다.
- (4) 오버플로우(Overflow)
연산 후 결과 값이 수의 표현범위를 벗어나는 경우를 오버플로우라고 하며 두수의 부호가 같은 경우에만 발생한다.
연산 후 부호 비트가 반전되면 오버플로우이다.

❖ 부동소수점 연산

실수 연산으로서 메모리에 기억된 형태가 지수 형태이므로 지수 형태에 의한 연산을 수행한다.

- (1) 덧셈과 뺄셈 : 부동소수점수의 덧셈과 뺄셈은 먼저 기수점의 위치를 조정하여야 한다. 이때 정확성을 위해 큰 지수 쪽에 맞춘다.
- (2) 곱셈 : 가수들은 곱하고 지수들은 더함으로써 이루어진다.
- (3) 나눗셈 : 지수의 뺄셈과 가수의 나눗셈으로 이루어진다.

주요포인트 꼭 알고 가야해!

주요포인트 꼭 알고 가야해!

(GO)

[04강] 제어장치

1. 컴퓨터시스템의 구성
2. 중앙처리장치의 구성
3. 제어장치의 정의
4. 레지스터
5. 명령어 형식
6. MAJOR STATE 1
7. MAJOR STATE 2
8. 기타주요명령

합격
비법대한민국 대표 IT강사
조 대 호

여러분의 인생드라마는
지금 잘 진행되시나요?
그 드라마의 주인공은
바로 당신입니다.
이 세상 모든사람들은
당신을 위한 조연이랍니다.
당신의 드라마가
성공한 그날!
당신의 감사메시지 속에서
저의 이름이 불려지기를
기원합니다.

정보처리기사/산업기사

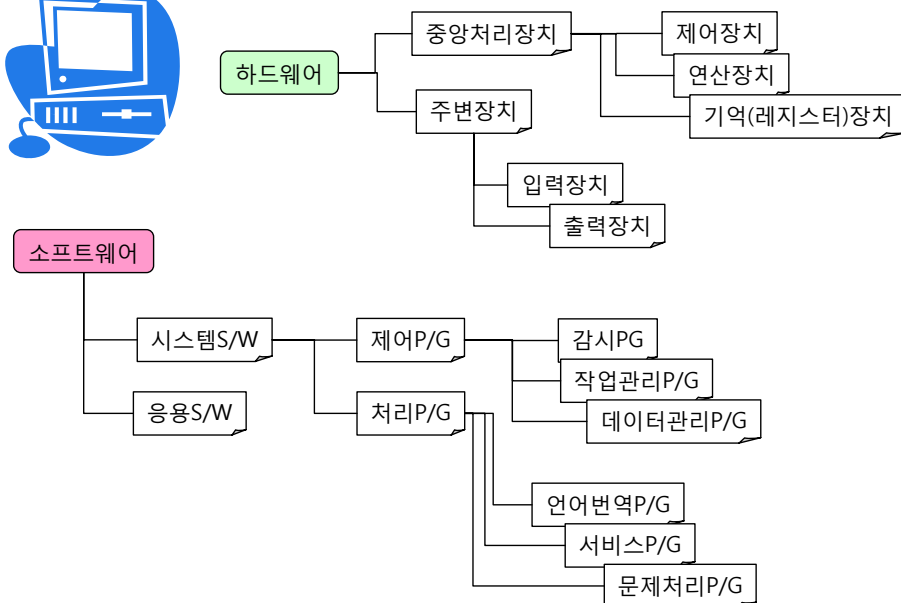
합격은 당연! 고득점 합격이 목표!

(GO)

1. 컴퓨터 시스템의 구성

JODAEHO

gisa plus



주요포인트 꼭 알고 가야해!

[동작원리에 의한 분류]

디지털 컴퓨터
아날로그 컴퓨터
하이브리드 컴퓨터

[사용 목적에 따른 분류]

범용 컴퓨터
특수용 컴퓨터

[규모에 따른 분류]

슈퍼 컴퓨터
메인 프레임 컴퓨터
미니 컴퓨터
마이크로 컴퓨터
개인용 컴퓨터



정보처리기사/산업기사

합격은 당연! 고득점 합격이 목표!

(GO)

[04강] 제어장치

Slides 18

❖제어장치(Control Unit)

- 컴퓨터 각 장치에 대한 지시, 감시, 감독.
- 주기억장치에서 읽어들이는 명령어 해독 -> 해당 장치에게 제어 신호를 전달하여 처리 지시.
- 명령해독기(Instruction Decoder): 명령 레지스터에 있는 명령어를 해독.
- 번지해독기 (Address Decoder): 명령 레지스터에 있는 명령어가 가지고 있는 번지 해독.
- 부호기(Encoder): 해독된 명령에 따라 각 장치로 보낼 제어 신호 생성.
- MAR(Memory Address Register. 메모리 주소 레지스터): 기억장치로 부터 입/출력되는 데이터의 주소 기억
- MBR(Memory Buffer Register. 메모리 버퍼 레지스터): 기억장치로 부터 입/출력되는 데이터 기억
- IR(Instruction Register. 명령 레지스터): 현재 수행 중인 명령어 기억
- PC(Program Counter. 프로그램 카운터(계수기)): 다음에 수행 할 명령어 기억

❖연산장치(ALU: Arithmetic Logic Unit)

- 사칙, 논리, 관계, 이동 연산
- 가산기(Adder) : 누산기와 데이터 레지스터의 값을 더하여 누산기에 저장.
- 보수기(Complementor): 뺄셈, 나눗셈 연산을 위해 보수로 바꾸어 가산.
- ACC(Accumulator. 누산기): 연산된 결과 일시 저장. 연산의 중심
- Data Register(데이터 레지스터): 연산에 사용될 데이터 일시 기억
- Status Register(상태 레지스터. SR 또는 PSR): CPU의 순간순간의 현재 상태 기록, 수행 순서제어(SR 또는 PSR)
- PSW(Program Status Word): SR에 저장된 값

❖기억장치

- 기억장치는 CPU에 의하여 실행될 프로그램이 저장되는 곳이며 주기억장치와 보조기억 장치로 나눈다

주요포인트 꼭 알고 가야해!

입력 장치

OMR(컴퓨터용 수성사인펜 이용. 시험답안 채점용)
OCR(빛 이용. 공공 요금 청구서)
MICR(자성체를 띤 특수 잉크. 수표, 어음)
키보드, 마우스, 라이트 팬, 스캐너, 디지털타이저

출력 장치

프린터, X-Y 플로터, COM

보조 기억 장치

입/출력 겸용 장치
HDD, CD-ROM, DVD, 플로피 디스크, 자기디스크, 자기테이프, 자기드럼

3. 제어장치의 정의

❖제어장치의 정의

- 기억된 프로그램(program)을 하나하나 불러내어 명령을 해독하는 장치
- 명령 코드가 명령을 수행할 수 있게 필요한 제어 기능을 제공해 주는 것
- 컴퓨터의 CPU 내부에서 주어진 명령어의 연산코드를 분석하여 레지스터, ALU, 기억장치, 버스 등에 각종 제어신호를 내보내는 장치
- 제어신호는 중앙 연산처리장치에서 마이크로 오퍼레이션이 순서적을 일어나게 한다.

❖제어장치의 기능

- 함수 연산 기능
- 자료 전달 기능
- 제어 기능
- 기억 기능

❖제어 방식의 종류

- Hard - Wired 방식(=논리회로 방식, 고정 배선 방식, H/W방식)
 - 처리속도가 빠르지만, 융통성이 부족하기에 비경제적.
 - 불규칙한 논리회로로 설계
- 마이크로 프로그램방식(=S/W방식)
 - 융통성이 좋아 경제적이지만 Hard - Wired 방식보다 느림
 - 마이크로프로그램은 전자계산기의 제작단계에서 제어 메모리(control storage) 속에 저장한다.
 - 마이크로프로그램은 마이크로 명령으로 형성되어 있다.

주요포인트 꼭 알고 가야해!

❖ 레지스터(Register)의 정의

- 중량 처리장치가 적은 양의 자료나 처리하는 동안의 중간 결과를 일시적으로 저장하기 위해 사용하는 고속의 기억 회로. 레지스터는 중앙 처리장치 내의 임시 기억장소로서 기억장치에서 읽은 값이나 쓸 값, 그리고 중간 계산 결과를 임시로 저장하는 역할을 한다.

❖ 레지스터의 종류

- Accumulator(ACC, 누산기)
 - 연산장치에 있는 레지스터의 하나로서 산술연산, 논리연산의 결과를 일시적으로 기억하는 장치
- Memory Buffer Register(MBR, 기억레지스터)
 - 주기억장치에 보낼 데이터나 주기억장치에서 보내온 데이터를 기억하는 레지스터
- Memory Address Register(MAR, 번지레지스터)
 - 기억장치 안에서 처리하는 데이터의 주소를 처리하는 레지스터
- Program Counter(PC, 프로그램카운터)
 - 프로그램의 수행순서를 제어하는 레지스터로, 다음에 실행할 명령의 번지를 기억
- Instruction Register(IR, 명령레지스터)
 - 현재 수행중인 명령의 내용을 기억
- PSW(Program Status Word Register, 상태레지스터)
 - 컴퓨터 시스템 내부에서 순간순간의 시스템 상태를 기록하고 있는 특별한 word로 CPU에서 수행되는 연산에 관련된 여러 가지 상태(Status) 정보를 기억하기 위해 사용됨
- Index Register(XR)
 - 실행 중에 오퍼랜드의 주소를 바꾸기 위해 그 내용이 사용되는 레지스터
- Base Register(BR)
 - 상대 주소지정에 사용되는 레지스터로 이 레지스터의 값과 명령에 지정된 offset을 더해 유효주소를 만든다.

주요포인트 꼭 알고 가야해!

[Register 전송방식]

- 직렬전송
 - 단일 회선에서 시프트 방식으로 1비트씩 차례대로 전송하는 방식
- 병렬 전송
 - 레지스터에 기억된 n개 비트를 한 번의 이동 명령에 의하여 전체가 한꺼번에 전송되게 하는 방식이다
- 버스 전송
 - 레지스터간의 공동선의 집합을 가지며 전송, 결선의 수를 줄이기 위해 사용

[General Purpose Register(GR)]

계산 결과의 임시 저장, 산술 및 논리 연산, 주소 인덱싱 등의 여러 가지 목적으로 사용될 수 있는 레지스터로, 주로 Acc, BR, XR로 사용됨

5. 명령어 형식

❖ 명령어 형식

- 컴퓨터가 어떻게 동작해야 하는지를 나타내는 것이며, 이것은 제어 장치에서 해독되어 그 동작이 이루어진다.

OP-Code(명령 코드부)	Mode	Operand(주소부)
-----------------	------	--------------

- Operation Code : 수행해야 할 동작을 명시한 부분
- Mode(1bit) : Operand의 주소 지정 방법을 나타내는 부분
- Operand : Operation Code가 수행해야 할 대상을 지정
- Operation code(동작부)
 - 이 크기는 해당 System에서의 사용 가능한 명령어의 개수에 의해 결정된다.
 - OP Code가 n bit인 경우 사용할 수 있는 명령어의 총 개수는 2^n
- Operand(자료부, 주소부)
 - Operand는 Operation이 처리할 데이터 또는 주소이다.
 - 주소인 경우는 Operand의 크기는 메모리 용량과 연관이 있다.
 - 메모리용량이 1M라면 필요한 주소 비트의 수는 20bit
 - MAR, PC = 주소 bit의 크기
 - MBR의 크기 = 워드의 크기

❖ Micro Operation

- 레지스터에 저장된 데이터를 가지고 실행되는 동작으로 하나의 클록 펄스(Clock Pulse) 동안 실행되는 기본동작을 말한다.
- 동기고정식 : 마이크로 동작 중에서 수행 시간이 가장 긴 것을 마이크로 주기 시간으로 정하는 방식
- 동기가변식 : 마이크로 동작들을 그 수행 시간이 유사한 것끼리 모아 한 개의 집단을 형성하여, 각 집단마다 서로 다른 마이크로 주기 시간을 주는 방식
- 비동기식 : 모든 마이크로 동작에 대해서 서로 다른 마이크로 주기 시간을 할당하는 방법

주요포인트 꼭 알고 가야해!

예1) 메모리 용량이 4096워드이고, 1워드는 16비트일 경우 PC, MAR, MBR의 크기를 구하라.

예2) 기억장치의 총 용량이 4096비트이고 워드 길이가 16비트일 때 프로그램 카운터(PC), 주소 레지스터(AR), 데이터 레지스터(DR)의 크기를 구하라.

6. Major State 1

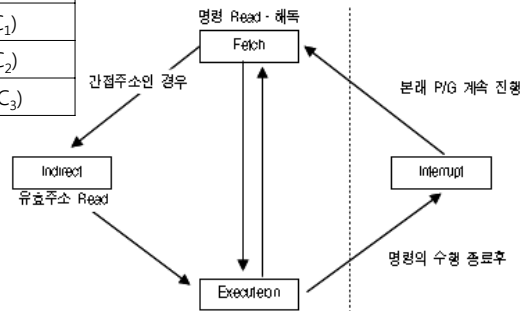
JODAEHO

gisa plus

❖ Major State

- CPU가 무엇을 하고 있는 지를 나타내는 register를 Major state register 라 한다. 종류는 다음과 같이 4가지가 있다.

F R	해당 cycle
0 0	Fetch Cycle(C_0)
0 1	Indirect Cycle(C_1)
1 0	Execute Cycle(C_2)
1 1	Interrupt Cycle(C_3)



❖ Fetch cycle(인출 주기)

- 기억장치에서 instruction을 읽어서 CPU로 가져오는 것으로 메이저 상태에서 인스트럭션의 종류에 대한 판단이 이루어지는 상태

C_0t_0	$MAR \leftarrow PC$	명령번지를 전송
C_0t_1	$MBR \leftarrow M(MAR), PC \leftarrow PC+1$	명령어를 읽고 MBR로 전송하고 다음 명령 준비
C_0t_2	$IR \leftarrow MBR$	명령 레지스터로 전송
C_0t_3	$R \leftarrow 1$	간접주기로 이동
C_0t_3	$F \leftarrow 1$	실행주기로 이동

정보처리기사/산업기사 합격은 당연! 고득점 합격이 목표!



[04강] 제어장치

Slides 23

주요포인트 꼭 알고 가야해!

7. Major State 2

JODAEHO

gisa plus

❖ Indirect cycle(간접 주기)

- 인스트럭션의 수행시 유효 주소를 구하기 위한 메이저 상태로 Operand가 간접 주소일 때 operand가 지정한 곳으로부터 유효 주소를 읽기 위해 기억장치에 한 번 더 접근함

C_1t_0	$MAR \leftarrow MBR(ADDR)$	명령어의 주소부분을 번지 해독기에 전송
C_1t_1	$MBR \leftarrow M(MAR)$	Operand의 번지를 읽는다
C_1t_2	NOP(No Operation)	
C_1t_3	$F \leftarrow 1, R \leftarrow 0$	실행주기로 이동

❖ Execution cycle(실행 주기)

- Fetch 상태나 Indirect 상태에서 구해진 유효번지에 자료를 읽어들이어 해당 명령들을 수행하는데, 이러한 동작을 Execute 상태라고 한다

C_2t_0	$MAR \leftarrow MBR(ADDR)$	명령어의 주소부분을 번지 해독기에 전송
C_2t_1	$MBR \leftarrow M$	Operand의 번지를 읽는다
C_2t_2	$AC \leftarrow AC$ (해당 operation) MBR	
C_2t_3	$F \leftarrow 0$	

❖ Interrupt cycle

- 현재 수행하고 있는 instruction의 수행을 마치고 발생된 상태를 처리하며 끝난 후 본래 프로그램을 계속할 수 있도록 하는 것

C_3t_0	$MBR \leftarrow PC, PC \leftarrow 0$	복귀주소를 저장하고 기억 위치 지정
C_3t_1	$MAR \leftarrow PC, PC \leftarrow PC+1$	기억될 위치 지정, interrupt로 분기할 번지 지정
C_3t_2	$M(MAR) \leftarrow MBR, IEN \leftarrow 0$	0번지에 복귀주소 기억, interrupt Enable을 불허
C_3t_3	$F \leftarrow 0, R \leftarrow 0$	Fetch Cycle로 이동

주요포인트 꼭 알고 가야해!

[인터럽트 허가(Interrupt Enable)]

컴퓨터의 CPU에 외부로부터 들어오는 인터럽트가 걸릴 수 있게 된 상태. CPU가 이상태가 아닐 때 들어오는 인터럽트는 무시되거나 보류된다.

정보처리기사/산업기사 합격은 당연! 고득점 합격이 목표!



[04강] 제어장치

Slides 24

❖AND(AND TO AC)명령

$q_0C_2t_0$	$MAR \leftarrow MBR(ADDR)$	유효주소를 전송
$q_0C_2t_1$	$MBR \leftarrow M(MAR)$	operand를 읽는다
$q_0C_2t_2$	$AC \leftarrow AC \text{ AND } MBR$	And 연산을 수행
C_2t_3	$F \leftarrow 0$	Fetch Cycle로 이동

❖ADD(ADD TO AC)명령

$q_1C_2t_0$	$MAR \leftarrow MBR(ADDR)$	유효주소를 전송
$q_1C_2t_1$	$MBR \leftarrow M(MAR)$	operand를 읽는다
$q_1C_2t_2$	$AC \leftarrow AC + MBR$	Add 연산을 수행
C_2t_3	$F \leftarrow 0$	Fetch Cycle로 이동

❖LDA(LOAD TO AC)명령

$q_2C_2t_0$	$MAR \leftarrow MBR(ADDR)$	유효주소를 전송
$q_2C_2t_1$	$MBR \leftarrow M(MAR), AC \leftarrow 0$	operand를 읽는다, AC를 Clear 시킨다.
$q_2C_2t_2$	$AC \leftarrow AC + MBR$	AC로 Load시킨다.
C_2t_3	$F \leftarrow 0$	Fetch Cycle로 이동

❖STA(STORE TO AC)명령

$q_3C_2t_0$	$MAR \leftarrow MBR(ADDR)$	유효주소를 전송
$q_3C_2t_1$	$MBR \leftarrow AC$	AC의 내용을 버퍼로 읽어온다.
$q_3C_2t_2$	$M(MAR) \leftarrow MBR$	메모리의내용을 전송
C_2t_3	$F \leftarrow 0$	Fetch Cycle로 이동

주요포인트 꼭 알고 가야해!

<BSA 명령>

서브루틴으로 분기하기 위해 사용되는 명령

$q_5C_2t_0$	$MAR \leftarrow MBR(ADDR)$ $MBR(ADDR) \leftarrow PC$ $PC \leftarrow MBR(ADDR)$
$q_5C_2t_1$	$M(MAR) \leftarrow MBR$
$q_5C_2t_2$	$PC \leftarrow PC + 1$
C_2t_3	$F \leftarrow 0$

주요포인트 꼭 알고 가야해!

GO

[05강] 주소지정과 기억장치

1. 주소지정방식
2. 주소지정에 따른 분류
3. 기억장치의 정의
4. 주기억장치
5. 자기테이프
6. 자기디스크
7. 기타기억장치

합격
비법대한민국 대표 IT강사
조 대 호

여러분의 인생드라마는
지금 잘 진행되시나요?
그 드라마의 주인공은
바로 당신입니다.
이 세상 모든사람들은
당신을 위한 조연이랍니다.
당신의 드라마가
성공한 그날!
당신의 감사메시지 속에서
저의 이름이 불려지기를
기원합니다.

❖ 주소지정방식의 분류

- Operand 개수에 따른 분류와 Operand의 주소지정방법에 따른 분류로 나눈다

❖ 3-Address Instruction

- 여러 개의 범용 레지스터를 가진 컴퓨터에 사용되며, 연산 후에 입력 자료가 변하지 않고 보존되는 인스트럭션의 형식으로 프로그램의 길이를 짧게 할 수 있다.

op-code	operand 1	operand 2	operand 3
---------	-----------	-----------	-----------

❖ 2-Address Instruction

- 계산 결과가 기억장치에 기억 될 뿐 아니라 중앙처리장치에도 남아 있어서 중앙처리장치 내에서 직접 시험이 가능하므로 시간이 절약된다.

op-code	operand 1	operand 2
---------	-----------	-----------

❖ 1-Address Instruction

- 연산의 처리 결과를 항상 누산기(Accumulator)에 저장하는 방식이다.

op-code	operand1
---------	----------

❖ 0-Address Instruction

- 모든 연산은 스택에 있는 자료를 이용하여 수행한다. 연산을 위한 스택을 갖고 있으며, 연산 수행 후에 결과를 스택(stack)에 보존한다. 명령어의 형식 가운데 연산에 사용된 모든 피 연산자 값을 상실한다.

주요포인트 꼭 알고 가야해!

[1-주소방식의 예]

LOAD A
ADD B
STORE C

2. 주소지정방식에 따른 분류

❖ 접근 방식에 따른 분류

- 묵시적 주소지정(Implied addressing mode)
- 즉시 주소지정(Immediate addressing mode)
 - 번지부에 표현된 값이 연산할 실제 데이터인 경우의 형식으로 data 접근이 가장 빠르며 메모리 접근횟수 = 0 으로 가장 적다.
- 직접 주소지정(Direct addressing mode)
 - 명령어의 주소부가 실제 데이터가 들어있는 주기억장치의 번지를 직접 갖고 있는 방식으로 메모리 접근횟수 = 1회 이다.
- 간접 주소지정(Indirect addressing mode)
 - 현재 번지부에 표현된 값이 실제 데이터가 기억된 번지가 아니고, 그 곳에 기억된 내용이 실제의 데이터 번지인 형식이며 메모리 접근횟수 = 최소 2회 이상이다
- 레지스터 모드
 - 범용 레지스터(R0 ~ R7)와 Acc와 데이터 전송하며 메모리 접근횟수 = 0 이다.
- 레지스터 간접 모드
 - 데이터 메모리의 어드레스를 범용 레지스터 R0, R1만을 사용하여 지정하며 메모리 접근횟수 = 1 이다.

❖ 계산에 의한 지정방식

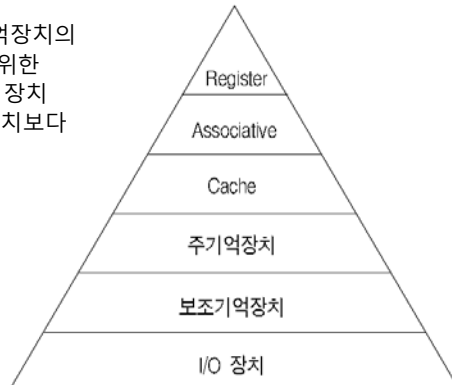
- 번지부에 표현된 값을 특정 값과 계산하여 상대적인 위치로 데이터를 지적하는 방법
- Indexed addressing mode : 명령의 번지 부분 + 인덱스 레지스터 => 유효번지
- Base addressing mode : 명령의 번지부분 + 베이스 레지스터 => 유효번지
- Relative addressing mode(상대 번지 지정) : 명령의 번지부분 + 프로그램 카운터(PC) => 유효번지

주요포인트 꼭 알고 가야해!

[간접주소를 사용하는 이유]
주소비트가 직접 지정 가능한 주소범위보다 더 넓은 범위의 주소지정이 가능
→ 주소범위 확장

❖기억장치의 정의

- 기억장치란 컴퓨터에서 처리할 프로그램이나 데이터를 기억하는 장치를 말하며 주기억장치와 보조기억장치로 구분된다.
- 주기억장치는 컴퓨터 내부에 존재하는 장치로 내부 기억장치라고도 하고 RAM과 ROM으로 구성되며 CPU에 의해 제어되는 장치로 즉시 필요한 프로그램이나 데이터가 기억된다.
- 보조기억장치는 주기억장치의 용량부족을 보조하기 위한 기억장치로 외부 기억 장치라고도 한다. 주기억장치보다 자료전송속도가 느린 반면 대용량이다.



❖ROM(Read Only Memory)

- 읽기전용 기억장치(read-only memory)의 준말로 반도체 기억장치의 하나로서 전원이 끊어져도 내용이 지워지지 않으며 읽을 수만 있는 비휘발성 메모리이다.
- ROM(masked ROM) - 제작 시 maker쪽에서 정보를 영구적으로 기록한 것
- PROM(Programmable ROM) - 사용자가 임의로 한번만 프로그램 가능
- EPROM(Erasable PROM) - 기억된 내용을 지우고 새로운 데이터를 기억시킬 수 있음
 - EEPROM(Electrically Erasable PROM) - 전기적인 방법
 - EAROM(Electrically Alterable ROM) - 자외선 또는 x선 이용

주요포인트 꼭 알고 가야해!

4. 주기억장치

❖RAM(=RWM : Read/Write Memory)

- random access memory(임의 액세스 기억장치)의 줄임말로써, 전원 공급이 끊기면 기억된 내용을 잃어버리는 휘발성 메모리이다. 각종 프로그램이나 운영체제 및 사용자가 작성한 문서 등이 불러와 작업할 수 있게 하는 장소로 모든 작업의 중심이 된다. 종류로는 SRAM과 DRAM이 있다.

❖SRAM과 DRAM의비교

구분	SRAM(Static Ram)	DRAM(Dynamic Ram)
사용범위	캐시 메모리	주기억장치
가격	비싸다	싸다
속도	빠르다	느리다
구조	복잡하다	단순하다
용량	작다	크다
재충전(Refresh)	불필요	필요

❖Magnetic core(자기 코어)

- 구성 : 4개의 도선으로 이루어져 있다.
 - x선, y선 - 정보를 기록하거나 읽을 때 사용. core를 자화하는데 사용하는 선
 - 금지선 (inhibit wire) - 코어의 자화를 소멸시키는 역할
 - 검출선 (sense wire) - 코어가 어느 방향으로 자화 되었는가를 검출하는 기능
- 특성
 - 파괴성 메모리이다.
 - 재 저장(store)을 해 주어야 한다.
 - 자화방향이 시계방향일 때 1을 기억한다(Flamming의 오른손의 법칙)

주요포인트 꼭 알고 가야해!

❖Magnetic tape(=SASD : Sequential Access Storage Device)

- 녹음테이프처럼 폴리에스테르 필름에 자화 물질을 도포하여 자화 형태로 자료를 사용할 수 있는 장치로 대용량의 저장매체로 순차처리만 가능한 매체이다.
- 자기테이프

IBG	논리 레코드	논리 레코드	논리 레코드	IBG	논리 레코드	논리 레코드	논리 레코드	IBG	...
-----	--------	--------	--------	-----	--------	--------	--------	-----	-----

- IBG(Inter Block Gap) : 블록과 블록 사이의 간격이다. 이 안에는 어떠한 내용도 들어 있지 않다.
- 논리 레코드(Record) : 항목의 집합으로 프로그램 내부의 처리 단위이며 하나의 레코드로 구성
- 물리 레코드(Block) : 입출력 단위인 블록으로 1개 또는 다수개의 논리 레코드로 구성
- 블록화 인수(Blocking Factor : B/F) : 하나의 물리 레코드(블록)을 구성하는 논리 레코드의 수 혹은 IBG와 IBG 사이에 있는 논리레코드 수

▪레코드 형식

- ① 고정 길이 비블럭 레코드(Unblocked Fixed Length Record)
- ② 고정 길이 블럭 레코드(Blocked Fixed Length Record)
- ③ 가변길이 비블럭 레코드(Unblocked Variable Length Record)
- ④ 가변길이 블럭 레코드(Blocked Variable Length Record)

▪자기테이프의 특징

- ① 순차처리만 가능.
- ② 대용량, 가격이 저렴해서 영구보존용으로 사용
- ③ 일괄처리에 적합

주요포인트 꼭 알고 가야해!

❖Magnetic disk(=DASD : Direct Access Storage Device)

- 대용량 보조기억장치로서 자기테이프 장치와는 달리 자료를 순차처리와 직접 처리할 수 있는 장치이다.

▪구성

- 트랙(Track) : 디스크상의 동심원을 의미
- 실린더(Cylinder) : 디스크의 중심축으로부터 동일한 거리에 위치하고 있는 트랙의 모임, 트랙의 수 = 실린더의 수
- 섹터(Sector) : 실제 데이터가 기록되는 위치로 트랙을 여러 구역으로 분리한 것
- 읽고/쓰기 헤드(Read/Write Head) : 디스크에 데이터를 읽고 쓰기 위한 장치
- 액세스 암 (Access Arm) : 데이터에 접근하기 위한 장치
- 디스크 팩(Disk Pack) : 여러 장의 디스크를 하나로 묶어 사용하는 것

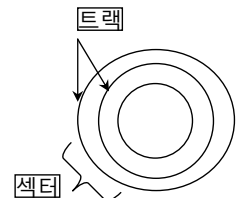
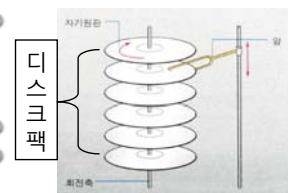
▪디스크 용량

- 디스크 팩에서의 사용 가능한 면 수
= 디스크 개수 × 2(면수) - 2(맨 윗면과 아랫면)
- 디스크에서의 사용 가능한 용량
= 트랙수 × 섹터수 × 섹터당 바이트 수 × 사용 가능한 면 수
- 예) 플로피 : 2Side × 80트랙 × 18섹터 × 512byte = 1474560byte = 1.44Mb

▪디스크 작동 시간

- Seek time(탐색시간) : 특정 트랙 위치까지 헤드가 도달하는 시간
- Search time(=Rotational delay, 검색시간, 회전지연시간) : 해당 트랙에서 자료가 있는 지정된 섹터를 찾는 데 까지 걸리는 시간
- Transfer time(전송시간) : 디스크로부터 데이터블록이 전송되는 시간
- Access Time(접근시간) : 정보를 기억장치에 기억시키거나 읽어내는 명령을 한 후 부터 실제로 정보를 기억 또는 읽기 시작할 때까지의 소요되는 시간
- Access time = Seek time + Search time + Transfer time

주요포인트 꼭 알고 가야해!



[Magnetic drum]

직경 20~30cm인 금속 원통의 표면에 자성물질을 입힌 것으로, 데이터를 액세스(Access)하는데 자기디스크나 자기테이프보다 속도가 빠르고 순차처리와 비순차처리 모두 가능하다.

❖ Buffer memory

- 저속 입출력 장치와 고속 내부 기억장치 간에 설치되어 여러 가지 기능을 수행하는 보조저장장치로서 한 장치에서 다른 장치로 Data를 송신할 때 일어나는 시간의 차이나 Data의 흐름 속도 차이를 보상하기 위해 만들어진 기억장치를 말한다.

❖ Cache memory

- 중앙 처리장치(CPU)의 속도와 주 기억장치의 속도차가 클 때 명령어(Instruction)의 수행 속도를 중앙 처리 장치의 속도와 비슷하도록 하기 위하여 사용하는 메모리
- CPU가 필요로 하는 데이터를 찾기 위해 캐시에 접근하여 데이터가 있으면 (Hit) 이를 읽어 들이고, 없으면(Miss) 주 기억장치로부터 읽어와 캐시로 전송한 후 캐시에서 읽어 들인다.

❖ Associative Memory(CAM, 내용지정메모리, 연상, 연관메모리)

- 기억장치에 기억된 정보를 액세스하기 위하여 주소를 사용하는 것이 아니고, 기억된 정보의 일부분을 이용하여 원하는 정보를 찾는 방법
- Parallel(병렬)로 접근하므로 접근시간이 빠르다.

❖ Memory Interleaving

- 한 Instruction을 위해 해독 또는 수행하는 동안에 다른 모듈에 있는 다음 Instruction을 접근할 수 있으며 기억장치를 여러 모듈로 구성하고 모듈간에 주소를 분배하여 번갈아 접근하는 방식으로 높은 시스템 처리 속도를 얻을 수 있다.
- 빠른 처리속도를 얻기 위해 중앙처리장치의 속도와 기억장치의 속도를 유효 Cycle 동안 병행 실행한다.

❖ 가상 메모리(Virtual memory)

- 사용자가 보조 메모리의 총 용량에 해당하는 기억 장소를 컴퓨터가 갖고 있는 것처럼 가상하고, 프로그램을 짤 수 있는 것을 말한다.
- 속도 향상이 아니라 주소 공간 확대가 목적이다

주요포인트 꼭 알고 가야해!

주요포인트 꼭 알고 가야해!



[06강] 입,출력장치

1. I/O장치
2. 입출력 방식
3. 채널
4. 데이터 전송방식
5. 인터럽트의 정의와 종류
6. 인터럽트의 우선순위
7. 병렬처리구조
8. Flynn에 의한 분류

합격
비법대한민국 대표 IT강사
조 대 호

여러분의 인생드라마는
지금 잘 진행되시나요?
그 드라마의 주인공은
바로 당신입니다.
이 세상 모든사람들은
당신을 위한 조연입니다.
당신의 드라마가
성공한 그날!
당신의 감사메시지 속에서
저의 이름이 불려지기를
기원합니다.

❖ I/O 장치의 정의

- CPU 내에서 수행할 데이터를 주기억장치 내에 기억시키는 입력 기능과 수행 결과를 주기억장치에서 출력 매체로 출력시키는 동작을 수행하는 장치.
- 입출력 모듈에 연결된 외부장치, 주변장치(Peripheral Device)

❖ 입출력에 필요한 하드웨어 기능

- 입출력 버스 : 주기억 장치와 입출력 장치 사이에서 데이터를 전달한다.
- 입출력 인터페이스 : 주기억 장치와 입출력 장치 사이의 속도와 단위 등의 차이를 해결하기 위한 장치
- 입출력 제어기 : 입출력 장치가 CPU로부터 자율적으로 동작하기 위한 별도의 입출력 제어 장치이다. 데이터 버퍼링, 제어 신호의 논리적 변환, 물리적 변환을 담당한다.

❖ CPU와 주변장치 비교

구분	CPU	주변장치
장치	전자장치	전기기계장치
동작속도	매우 빠르다	느리다
정보단위	워드 단위	문자단위
오류율	낮다	높다

❖ 입출력 장치의 종류

구분	종 류
입력 장치	카드 판독기, OMR, OCR, 키보드, 마우스, 스캐너, 라이트 펜 등
출력 장치	프린터, 모니터, X-Y 플로터 등
입출력장치	자기테이프, 자기드럼, 자기디스크, 플로피 디스크 등

주요포인트 꼭 알고 가야해!

2. 입출력 방식

❖ 프로그램에 의한 I/O

- 프로그램이 입출력 동작을 시작, 지시 종료시킨다.
- 데이터전송은 입출력버스를 통하여 프로세서와 주변장치 사이에서 이루어진다.
- 데이터 전송 시에 CPU가 주변장치를 계속해서 감시하는 폴링(Polling) 방식에 의해서 수행되기 때문에 저속이고, 시간 낭비가 발생한다.

❖ 인터럽트에 의한 I/O

- 프로그램 명령 중 입출력 명령이 있으면 인터럽트를 발생시켜 입출력 동작의 개시를 지시하고, CPU는 다른 일을 한다.
- 프로그램에 의한 방법에서는 CPU가 인터페이스에게 현재의 상태가 BUSY인지 DONE 인지 계속 묻는데 인터럽트는 CPU가 입출력 개시를 지시한 후 더 이상 간섭하지 않도록 한다.

❖ DMA(Direct Memory Access)에 의한 I/O

- DMA 방식은 중앙 처리 장치로부터 입출력 지시를 받으면 직접 주 기억장치에 접근하여 데이터를 꺼내어 출력하거나 입력한 데이터를 기억시킬 수 있고, 입출력에 관한 모든 동작을 자율적으로 수행한다.
- CPU 개입 없이 주변장치와 주기억장치의 데이터 전송이 이루어지는 방법.
- 자기 테이프나 자기 디스크와 같은 대용량의 고속 I/O 장치에 적합하다.
- DMA는 그 작동 도중 시스템의 버스를 사용하게 되는데 이 때 CPU와 DMA가 동시에 버스를 요청했을 때 DMA에 우선권이 주어지는 것을 사이클 도용(cycle stealing)이라 한다.

❖ 사이클 스틸링(Cycle Stealing)

- DMA 제어기가 한번에 한 데이터 워드를 전송하고 버스의 제어를 CPU에게 돌려주는 것
- 중앙처리장치와 입/출력장치가 동시에 주 기억장치를 접근하려고 하는 경우, 입/출력장치에 우선순위를 부여하여 주 기억장치를 액세스 하는 동안 중앙처리장치에서 주 기억장치에 대한 접근 작업을 양보하도록 하는 기법

주요포인트 꼭 알고 가야해!

❖정의

- 중앙 처리 기능을 가진 소형 처리기를 DMA 위치에 두고 입출력에 관한 제어 사항을 전담하도록 하는 특수 컴퓨터를 채널(channel) 이라고 한다.
- CPU 간섭 없이 입출력 동작을 수행하도록 지시하고, 작업이 끝나면 CPU에게 인터럽트로 알린다.

❖특징

- 입출력 명령 해독
- 각 입출력장치에 입출력 명령 지시 및 명령의 실행을 제어
- 입·출력은 DMA방법으로도 수행한다.
- 입·출력은 수행 중 어떤 오류조건에서 CPU에 인터럽트를 걸 수 있다.
- 자체적으로 자료의 수정 또는 코드 변환 등의 기능을 수행할 수 있다.

❖채널의 종류

- Selector channel
 - 어느 하나의 I/O장치에 전용인 것처럼 사용하여 자료를 전송
 - 고속의 I/O장치에 사용
 - ex) 자기 테이프, 자기 디스크, 자기드럼
- Multiplexer channel
 - 시분할적으로 동시에 다수의 I/O장치를 사용하여 자료를 전송하는데 사용
 - 저속의 I/O장치에 사용
 - ex) 카드 리더, 라인 프린터

❖스풀링(Spooling)

- 효율적인 입출력을 위하여 고속의 CPU와 저속의 입출력장치가 동시에 독립적으로 동작하게 하여 아주 높은 효율로 여러 작업을 병행 작업할 수 있도록 해줌으로써 다중 프로그래밍 시스템의 성능 향상을 가져올 수 있게 하는 방법

주요포인트 꼭 알고 가야해!

[버퍼링(Buffering)]

- 저속의 입출력 장치와 고속의 CPU간의 속도차이를 해소하기 위해서 나온 방법
- 한 레코드를 읽어서 CPU가 그것에 대한 작업을 시작함과 동시에 입/출력 장치가 필요한 레코드를 미리 읽어 CPU에 저장해 둔다.
- 버퍼링은 CPU와 I/O 장치를 항상 바쁘게 하여 I/O 장치의 느린 속도를 보완하는 방법이다.
- 주기억장치의 일부를 버퍼로 사용

4. 데이터전송방식

❖비동기 데이터 전송

- 두개의 독립적인 장치 간에 데이터 전송은 두 장치가 서로 다른 클럭을 사용하기 때문에 비동기 데이터 전송을 사용한다.
- 두 독립적인 장치 사이의 비동기적인 데이터전송을 이루기 위해서는 데이터가 전송될 시각을 알리기 위해 두 장치 사이에 제어 신호가 교환되어야 한다.
- 이러한 제어신호교환 방법에는 아래와 같이 스트로브 펄스와 핸드셰이킹 기법이 있다.

❖스트로브 제어

- 입출력 장치는 중앙 처리 장치와 속도차가 많이 나기 때문에 중앙 처리 장치와는 동기화될 수 없으며, 시간 펄스를 자체적으로 가지고 있어 자율적 동작이 되도록 한다.
- 스트로브 신호 : 비동기적인 두 개 이상의 장치 사이에서 데이터를 전송하는 시점에 데이터 전송을 알려주기 위한 신호. 별도의 회선 필요하다.
- 단점 : 전송한 데이터를 수신 쪽에서 확실하게 수신하였는지를 알 수 없다.

❖핸드셰이킹 제어

- 송신 쪽에서 데이터를 버스에 실은 후 제어선으로 수신 쪽에 데이터가 송신되고 있음을 알려주고, 수신 쪽에서는 데이터를 수신한 후 수신이 완료되었음을 송신 쪽에 알려주므로, 이 방법은 데이터가 정확히 전달되었는지를 알 수 있다.
- 송신 쪽과 수신 쪽이 동시에 동작해야 하므로 높은 신뢰성을 가진다.
- 핸드셰이킹이란?
 - 전기적으로 연결된 두 장치 사이에서 데이터를 교환할 때 동기를 맞추기 위해 일련의 신호들을 주고받는 절차
 - 송신측과 수신측 양쪽에서 동시에 제어신호 전송
 - 별도의 회선 필요

주요포인트 꼭 알고 가야해!

❖ 인터럽트의 정의

- 전자계산기에서 어떤 특수한 상태가 발생하면 그것이 원인이 되어 현재 실행하고 있는 프로그램이 일시 중단되고, 그 특수한 상태를 처리하는 프로그램으로 옮겨져 처리한 후 다시 원래의 프로그램을 처리하는 현상을 말한다.

❖ 인터럽트 종류

- ① 외부 인터럽트 (External Interrupt)
조작원이 콘솔에서 인터럽트 스위치를 눌러서 강제로 인터럽트 시키거나 정해놓은 시간이 되었을 때 계시기(timer)에서 인터럽트를 발생시키는 경우 및 전원 스위치가 꺼지거나 정전이 되는 경우에 해당한다.
- ② 기계오류 인터럽트(Machine Check Interrupt)
프로그램이 실행되는 중에 어떤 장치의 고장으로 인하여 제어 프로그램에게 조치를 취해줄도록 요청하는 인터럽트이다.
- ③ 프로그램 검사 인터럽트(Trap, Internal, Program Check Interrupt)
0으로 나누는 연산 및 오버플로(Overflow)나 언더플로(UnderFlow)가 발생했을 때 또는 프로그램 실행 중에 잘못된 데이터를 사용하거나 보호된 구역에 불법 접근하는 등의 프로그램 자체에서 잘못되었기 때문에 발생하는 인터럽트이다.
- ④ 입출력 인터럽트(I/O Interrupt)
입력이나 출력 명령을 만나면 현재 프로그램의 진행을 정지하고 입출력을 담당하는 채널 같은 기구에 입출력이 이루어지도록 명령하고 중앙 처리 장치는 다른 프로그램을 실행하도록 하는 인터럽트이다.
- ⑤ 감시 프로그램 호출 인터럽트(SVC, Supervisor Call Interrupt)
프로그램 내에서 제어 프로그램에게 인터럽트를 요청하는 명령으로 인터럽트 시키는 방법이다. 입/출력 수행, 기억장치 할당 및 오퍼레이터와 대화 등을 하기 위해 발생하는 인터럽트이다.

주요포인트 꼭 알고 가야해!

[인터럽트의 발생요인]

- ① 정전이나 기계적 고장
- ② 프로그램의 잘못
- ③ 조작원의 지시
- ④ 입출력 기능
- ⑤ 프로그램에서 인터럽트 요청

7. 인터럽트 우선순위

❖ 우선순위 파악하는 방법

- (1) 하드웨어적인 방법(Daisy chaining)
인터럽트 라인을 직렬로 연결하여 우선순위를 결정하는 방법으로 우선순위가 높은 장치를 선두에 두고 우선순위에 따라 직렬로 연결하는 방법이다.
응답속도가 빠르기 때문에 주로 사용하나 비경제적이고 유연성이 적다.
- (2) 소프트웨어적인 방법(Polling)
인터럽트 요청신호 플래그를 차례로 검사하여 인터럽트의 원인을 판별하는 방식으로 H/W적인 방법에 비해 경제적이고 유연성이 좋으나 응답속도가 느리다.

※ 인터럽트 우선순위

정전 인터럽트 > 기계고장 인터럽트 > 외부 인터럽트 > I/O 인터럽트 > 프로그램 체크 인터럽트 > SVC 인터럽트

❖ 인터럽트 발생시 처리순서

- ① CPU에 인터럽트 요청
- ② 현재 작업 중인 주소를 메모리에 저장
- ③ 인터럽트 인지신호 발생
- ④ 벡터 인터럽트 처리
- ⑤ 리턴에 의한 복귀

❖ 인터럽트 벡터

- 각 인터럽트 신호에 대해 그것이 들어오면 분기할 인터럽트 처리루틴의 시작 주소를 기억하고 있는 기억장소 또는 그 루틴의 주소.
- 인터럽트가 컴퓨터에서 발생하였을 때 프로세서의 인터럽트 서비스가 특정의 장소로 점프하도록 되어 있는 것

주요포인트 꼭 알고 가야해!

❖ 병렬처리구조

- 병렬의 개념은 폰노이만 형식의 컴퓨터가 갖는 문제점을 개선하고 보다 많은 데이터 처리와 고속처리를 위해 발전된 개념이다.

❖ 병렬처리에서 발생하는 문제점

- 분할문제
 - 하나의 순차성을 갖는 프로그램을 최대의 병렬성을 갖도록 단계별로 분할할 때 Task의 단위를 어떻게 할 것인가하는 문제
- 스케줄링의 문제
 - 분할된 태스크들을 실제 실행하기 위하여 스케줄링상에 문제가 발생하게 된다.
 - 정적스케줄링 : 각 태스크의 프로세서 할당과 수행순서가 컴파일시에 컴파일러에 의해 결정되는 스케줄링
 - 동적 스케줄링 : 수행시에 태스크들을 프로세서에서 할당하는 방식
- 동기화문제
 - 공유된 데이터가 한 프로세서에 의해서 변경되지 못하도록 하기위한 동기화 작업이 필요한데 특정한 프로세서만을 액세스하지 못하도록 처리순서를 정하는 것.
 - 버스잠금방식, 플래그기법, 세마포어기법
- 캐시메모리 관련문제
 - 여러 개의 프로세서들이 하나의 버스를 통하여 메모리를 공유함으로써 메모리 내의 데이터 충돌이 발생되며 속도가 떨어지게 되는데 캐시메모리를 사용함으로써 버스의 충돌을 줄이고 속도를 증가시킬 수 있다.

❖ 병렬처리기의 정의와 종류

- 정의 : 연산이나 처리의 고속화를 위해 연속 일괄처리를 분할하여 동시에 처리하는 것
- Pipeline 기법 : 한번에 여러기기에 명령어가 실행단계에 있게하여 성능을 향상시킴
- 벡터 프로세서 : 연산을 빠르고 효율적으로 수행하도록 구성
- 배열 프로세서 : 데이터를 고속으로 처리하기 위해 연산장치를 병렬로 배열한 처리구조
- 데이터 흐름 컴퓨터 : 요구되는 데이터가 이용 가능한 명령어를 실행시키는 구조

주요포인트 꼭 알고 가야해!

9. Flynn에 의한 분류

❖ SISD (Single Instruction stream Single Data stream)

- 일반적인 컴퓨터구조(폰노이만 형)가 여기에 속하며 단일 명령흐름에 대한 단일 데이터 흐름이 특징이다.
- 단일처리, 생산성 ↓, 시간적인 병렬처리를 할 수 있다.

❖ SIMD (Single Instruction stream Multiple Data stream)

- 단일 명령 흐름에 대한 다중 데이터 흐름이 특징이다.
- 다중처리, Multiprogramming, 배열처리기에서 주로 사용.

❖ MISD (Multiple Instruction stream Single Data stream)

- 다중명령흐름에 대한 단일 데이터 흐름이 특징이다.
- 이론적으로만 존재하며, 현재 사용하지 않는다.

❖ MIMD (Multiple Instruction stream Multiple Data stream)

- 다중명령흐름에 대한 다중 데이터흐름이 특징이다.
- 진정한 의미의 병렬 처리구조로 대부분 약결합 형태의 구조적 특징을 가지나 강결합 형태도 있다.

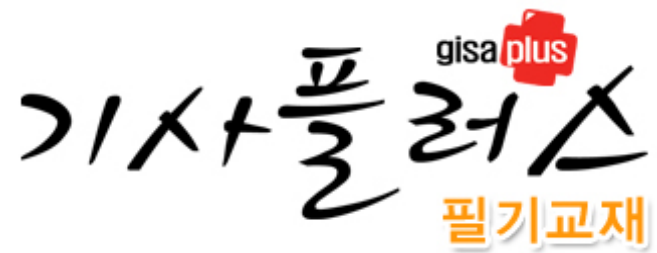
❖ 다중처리기

- 다중처리기란 시스템상의 여러 프로세서들에게 여러 개의 독립적인 작업을 각각 배정하여 하나의 프로세서는 하나의 작업만을 수행하도록 하는 다중처리를 위하여 두 개 이상의 프로세서를 동시에 수행할 수 있도록 여러 프로세서를 갖춘 컴퓨터 시스템.

❖ 약결합과 강결합

- 약결합 : 독립된 기억장치와 운영체제로 프로세서 상호간의 정보를 교환하는 방식
- 강결합 : 하나의 공유 기억장치와 운영체제를 갖고 프로세서 간 정보를 교환하는 방식

주요포인트 꼭 알고 가야해!



정보처리기사/산업기사 **필기**

운 영 체 제



주요포인트 꼭 알고 가야해!



[01강] 운영체제의 개요

1. 운영체제의 개념
2. 운영체제의 종류
3. 운영체제의 구성
4. 프로그래밍 언어와 번역과정
5. 매크로와 부 프로그램



대한민국 대표 IT강사
조 대 호

여러분의 인생드라마는
지금 잘 진행되시나요?
그 드라마의 주인공은
바로 당신입니다.
이 세상 모든사람들은
당신을 위한 조연이랍니다.
당신의 드라마가
성공한 그날!
당신의 감사메시지 속에서
저의 이름이 불려지기를
기원합니다.

정보처리기사/산업기사 합격은 당연! 고득점 합격이 목표! (GO)

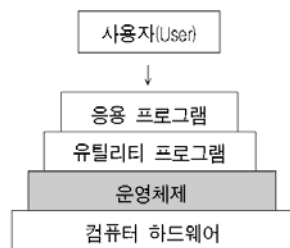
1. 운영체제의 개념

JODAEHO

gisa plus

❖ 운영체제(Operating system)의 정의

- 제한된 컴퓨터 시스템의 자원을 보다 효율적으로 관리하고 운영함으로써 사용자들에게 편의성을 제공하고자 하는 컴퓨터 하드웨어와 사용자간의 매개체 역할을 하는 시스템 프로그램이다. 운영체제는 컴퓨터 자원들인 기억장치, 프로세서, 파일 및 정보, 네트워크 및 보호 등을 효율적으로 관리할 수 있는 프로그램의 집합으로 주목적은 컴퓨터 시스템을 편리하게 이용할 수 있게 하는데 있다.



❖ 운영체제의 기능

- 사용자와 컴퓨터 시스템간의 인터페이스 기능 제공
- 사용자간의 하드웨어를 공동으로 사용 허용
- 사용자간의 데이터를 공유하며 사용자간의 자원 스케줄링
- 입출력 보조 역할과 컴퓨터 시스템의 에러 처리

❖ 운영체제의 목적

- 사용자 인터페이스의 제공
- 주변장치 관리
- 자원 스케줄링
- 신뢰성의 향상

주요포인트 꼭 알고 가야해!

[운영체제의 성능평가 요인]

1. 응답시간(Turn Around Time)단축
2. 처리능력(Throughput) 향상
3. 신뢰도(Reliability)향상
4. 사용가능도(Availability)증대



정보처리기사/산업기사 합격은 당연! 고득점 합격이 목표!



[01강] 운영체제의 개요

Slides 2

❖ 일괄 처리 시스템(Batch Processing System)

- 처리할 자료를 즉각 처리하는 것이 아니라 일정기간 또는 일정량을 모아서 한 번에 처리하는 방식
- 장점 : 컴퓨터 시스템을 효율적으로 사용 가능
- 단점 : 작업시간이 길어서 반환시간(Turn Around Time)이 길어진다

❖ 다중 프로그래밍 시스템(Multi-Programming System)

- 하나의 CPU로 여러 개의 프로그램을 동시에 실행시키는 방식
- 장점 : CPU의 유휴시간과 반환시간을 줄일 수 있다.

❖ 시분할 체제(Time Sharing System)

- 사용자는 단말장치를 이용하여 운영체제와 상호 작용하며, 시스템은 일정 시간 단위로 CPU를 한 사용자에서 다음 사용자로 신속하게 전환함으로써, 각각의 사용자들은 실제로 자신만이 컴퓨터를 사용하고 있는 것처럼 사용할 수 있는 처리 방식

❖ 실시간 시스템(Real-Time System)

- 사용자의 요구에 즉각 응답하는 방식으로 공장제어, 비행기 제어 시스템, 은행 온라인 업무 등에서 사용되는 시스템
- 장점 : 처리시간을 단축할 수 있으며 비용이 절감되고, 자료발생 후 직접 입출력됨.
- 단점 : 자료가 무작위로 도착하므로 입출력 자료의 일시 저장 및 대기가 필요함.

❖ 분산처리 시스템(Distributed Processing System)

- 지역적으로 분산된 여러 개의 컴퓨터를 연결해서 일을 분담 처리하는 시스템.
- 지리적으로는 떨어져 있으나 통신망으로 연결되어 있어 마치 하나의 컴퓨터 시스템처럼 동작하는 분산 시스템에서 작업이 처리되는 것.
- 장점 : 자원의 공유, 계산 속도의 증가, 신뢰성 향상
- 단점 : 구축이 어렵다, 비용이 많이 든다, 보안 유지가 어렵다

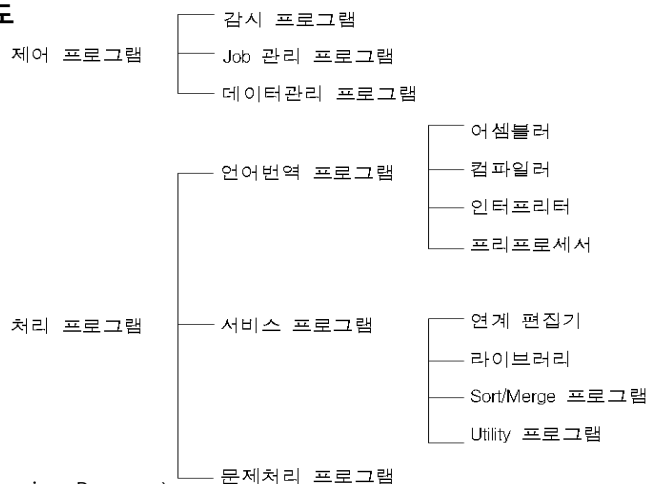
주요포인트 꼭 알고 가야해!

[병렬 처리 시스템(Parallel Processing System)]

입출력 채널 또는 처리기와 같은 장치에 둘 이상의 프로세스를 동시에 수행하는 것으로 병렬처리의 목적은 연산속도를 높임으로써 처리능력을 높이는 것임

3. 운영체제의 구성

❖ 운영체제의 구성도



❖ 제어 프로그램

- 감시 프로그램(Supervisor Program)
 - 제어 프로그램의 중추적 기능을 담당하는 부분.
 - 처리 프로그램의 실행과 시스템 전체의 동작 상태를 감독하는 기능.
- 데이터 관리 프로그램(Data Management Program)
 - 시스템에서 취급하는 각종 File과 데이터를 처리.
 - 주기억장치와 보조기억장치사이의 자료전송, File의 조작 및 사용, 입출력 데이터와 프로그램과의 논리적 연결을 담당.
- Job 관리 프로그램(Job Management Program)
 - 업무 처리에 대한 자동적인 이행을 위한 준비 등의 기능을 수행
 - Job의 연속적인 스케줄링, 시스템 자원의 할당 등을 담당.

주요포인트 꼭 알고 가야해!

[언어번역 프로그램]

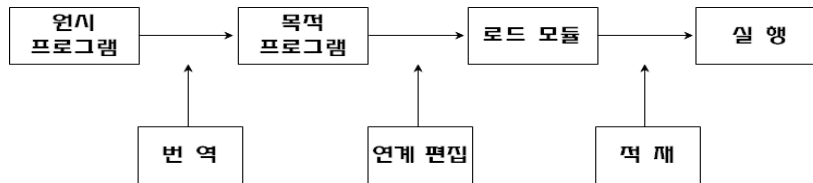
- ① 어셈블러(Assembler)
어셈블리어로 작성된 프로그램을 기계어로 번역해 주는 언어 번역 프로그램
- ② 컴파일러(Compiler)
고급언어로 작성된 프로그램을 저급언어로 번역해 주는 프로그램으로 목적 프로그램을 생성한다. 한번 번역한 후에는 다시 번역할 필요가 없으므로 실행 속도가 빠르다.
- ③ 인터프리터(Interpreter)
고급언어로 작성된 프로그램을 번역 즉시 바로 실행하며, 목적 프로그램을 생성하지 않는다. 실행할 때 마다 번역하므로 실행 속도가 느리다.



❖프로그래밍 언어의 종류

- 저급언어 : 기계중심 언어, 처리속도가 빠르다, 호환성(이식성)이 나쁘다(기계어, 어셈블리어)
- 고급언어 : 인간중심 언어, 처리속도가 느리다. 호환성이 좋다, Basic, C, Pascal 등등

❖언어의 번역과정



❖용어

- 원시프로그램(Source Program) : 사용자가 작성한 처음(원본)의 프로그램
- 목적 프로그램(Object Program) : 컴파일러에 의해 기계어로 번역된 프로그램
- 연계편집(Linker) : 목적 프로그램을 실행 가능한 프로그램으로 만드는 과정
- 로드모듈(Load Module) : 목적 모듈에 번지 등을 지정하여 실행할 수 있는 형태의 프로그램
- 적재(Loader) : 프로그램을 실행시키기 위해 주기억 장치에 적재하는 프로그램
 - 할당(Allocation) : 주기억 장치 안에 빈 공간을 할당
 - 연결(Linking) : 목적 모듈들 사이의 기호적 외부 참조를 실제적 주소로 변환
 - 재배치(Relocation) : 종속적인 모든 주소를 할당된 주기억 장치 주소와 일치하도록 조정
 - 적재(Load) : 기계 명령어와 자료를 기억 장소에 물리적으로 배치

주요포인트 꼭 알고 가야해!

[로더의 기능별 주체]

할당 : 프로그래머
연결 : 프로그래머
재배치 : 어셈블러
적재 : 로더



5. 매크로와 부프로그램

❖매크로(Macro)의 특징

- Open Subprogram이라고도 하며 자주 반복되는 일련의 명령을 하나의 블록으로 묶어 이름을 부여하고, 사용할 곳에 지정한 이름만 적어 수행함
- 프로그래머가 어셈블리 언어로 프로그램을 작성할때 반복되는 일들을 효과적으로 처리하기 위해 사용한다.
- 번역되기전에 매크로 처리기에 의해 확장이 일어난다.
- 실행이 빠르지만 메모리 사용량이 많아진다.
- 매크로를 사용하면 일반적으로 프로그램의 크기가 커짐

❖부 프로그램(Sub Program)의 특징

- 주 프로그램 또는 다른 부프로그램에 의해 사용되는 독립적인 단위.
- 한 프로그램 내에서 특정한 일이 여러 번 실행될때 사용한다.
- 컴파일 후에 Call과 Return 에 의해 실행된다.
- 복귀 주소의 저장과 조회는 스택구조를 이용한다.
- 부 프로그램을 사용하면 프로그램의 크기를 상대적으로 줄일 수 있음
- 프로그램의 크기가 작아져서 기억장소가 절약된다.
- 가독성이 높고 유지보수가 용이 하다.
- 실행시간은 상대적으로 느리다.

❖매크로와 부프로그램 비교

- 매크로 : 컴파일 전에 확장이 일어나며 실행은 빠르지만 메모리 사용량이 많음
- 부 프로그램 : 컴파일 후 실행이 이루어 지며 실행이 상대적으로 느리지만 메모리 사용량이 상대적으로 적음

주요포인트 꼭 알고 가야해!

[매크로 프로세서]

매크로 명령문을 해당 어셈블리 언어의 명령문으로 확장해 주는 시스템 프로그램

[매크로 처리 과정]

매크로 정의인식
매크로 정의저장
매크로 호출인식
매크로 호출확장



주요포인트 꼭 알고 가야해!

 [02강] 프로세스

1. 프로세스 관리
2. 스케줄링
3. 스케줄링 관련 문제

**합격
비밀**



대한민국 대표 IT강사
조 대 호

여러분의 인생드라마는
지금 잘 진행되시나요?
그 드라마의 주인공은
바로 당신입니다.
이 세상 모든사람들은
당신을 위한 조연이랍니다.
당신의 드라마가
성공한 그날!
당신의 가슴메시지 속에서
저의 이름이 불려지기를
기원합니다.

정보처리기사/산업기사

합격은 당연! 고득점 합격이 목표!

(GO

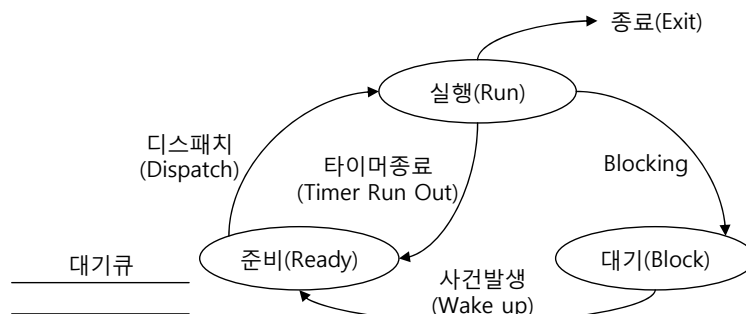
1. 프로세스 관리

JODAEHO

gisa plus

❖ 프로세스(Process)의 정의와 상태

- 정의 : 현재 실행 중인 프로그램, PCB를 가진 프로그램, 프로세서가 할당되는 실체
- 프로세스 상태
 - 준비(Ready)상태 : CPU를 사용할 수 있도록 실행 상태로 전이하는 처리과정
 - 실행(Run)상태 : 프로세스가 프로세서를 차지하고 있어 실행중인 상태
 - 대기(Block)상태 : 프로세서가 어떤 사건이 일어나기를 기다리는 상태



❖프로세스 제어 블록(PCB; Process Control Block)

- 프로세스에 대한 중요한 정보를 포함하고 있는 자료 구조
- 프로세서의 현 상태, 프로세스의 고유한 식별자, 프로세스의 우선순위, 프로세스에 할당된 메모리에 대한 포인터, 프로그램카운터, CPU 레지스터 등에 관한 정보의 내용을 포함

주요포인트 꼭 알고 가야해!

- 디스패치 (Dispatch)

준비상태에서 대기하고 있는 프로세스 중 하나가 스케줄링되어 중앙처리장치를 할당받아 실행상태로 전이되는 과정

- Timer Runout

프로세스가 할당된 시간 (Time Slice)이 지났는데도 CPU의 점유권을 반환하지 않으면 인터럽트를 발생하여 프로세서의 제어권을 강제로 운영체제에게 인수하고 해당 프로세스는 준비상태로 전이한다.

- Wake-up(I/O completion or event completion)

입출력 작업이 끝난 프로세스는 대기상태에서 준비상태로 전이된다.

정보처리기사/신입기사

합격은 당연! 고득점 합격이 목표!



[02강] 프로세스

Slides 8

❖스케줄링의 개념

- 컴퓨터 시스템의 자원을 효율적으로 사용하기 위해 작업 순서와 시간을 할당하는 것

❖선점형(Preemptive) 스케줄링 기법

- 하나의 프로세스가 CPU를 점유하고 있을 때 작업중인 프로세스 작업을 중단시키고 CPU를 점유하는 방식
- 라운드 로빈(Round Robin) : 각 작업은 주어진 시간(시간할당량)동안만 작업함
 - 시간 할당량과의 관계
 - 시간 할당량이 너무 클 경우 FIFO와 같아져 비선점 형식이 된다.
 - 시간 할당량이 작을 경우에는 작업 교환에 따른 오버헤드가 커져 비효율적
- SRT(Shortest Remaining Time) : 남아있는 수행 시간이 가장 짧은 작업을 먼저 처리함
- 다단계 피드백 큐(MFQ) : 다양한 특성의 작업이 혼합된 경우 매우 유용한 방법

❖비선점형(Non-Preemptive) 스케줄링

- 프로세서가 CPU를 차지하고 있을때는 점유할 수 없고 작업이 끝날 때까지 CPU를 독점하는 방식
- FIFO(First In First Out) : 먼저 들어온 것을 먼저 처리함
- 우선순위 스케줄링 : 우선순위가 높은 작업을 먼저처리함(기아상태가 발생함)
- SJF(Shortest Job First) : 수행 시간이 가장 짧은 작업을 먼저 처리함
- HRN(Highest Response-ratio Next)
 - 우선순위를 계산해서 우선순위가 높은(짧은)작업을 낮은(긴) 작업보다 먼저 처리
 - 우선순위 계산공식 = (대기시간+서비스 받을 시간) / 서비스 받을 시간
- Deadline(기한부)

주요포인트 꼭 알고 가야해!

[Aging 기법]

프로세스의 우선순위를 프로세스가 자원을 기다린 시간에 비례하여 부여하는 기법으로 프로세스의 무기한 기다림을 방지할 수 있다. 즉, 대기 중인 프로세스의 우선순위가 새로 도착하는 프로세스의 우선순위보다 높아져 처리를 받을 수 있게 된다.



3. 스케줄링관련 문제

문제1) Job의 도착 시간 및 실행시간이 다음과 같다면 FIFO, SJF, Round Robin 정책을 사용할 경우 평균반환시간은 얼마인가?

- FIFO : $(10+39+42+49+61)/5=40.2$
- SJF : $(3+10+20+32+61)/5=25.2$
- RR : Time Quantum = 10
 $(10+20+23+30+40+30+12+11)/5=35.2$

JOB					
대기					
실행					

작업	버스트시간
1	10
2	29
3	3
4	7
5	12

문제2) 다음 표와 같이 작업이 제출되었을 때, SJF 정책을 사용하여 스케줄링하면 평균 turnaround 시간은 얼마인가?

작업	제출시간	실행시간
Job1	0	4
Job2	1	6
Job3	2	3

JOB			
대기			
실행			
반환			

문제3) Job의 도착 시간 및 실행시간이 다음과 같다면 Round Robin정책을 사용할 경우 average turnaround time은 약 얼마인가? (단, quantum = 10초)

JOB번호	예상 실행 시간(단위 : 초)	도착 시각
1	10	10시 00분
2	29	10시 00분 10초
3	3	10시 00분 15초
4	7	10시 00분 16초
5	12	10시 00분 20초

주요포인트 꼭 알고 가야해!

반환시간=대기시간+실행시간

주요포인트 꼭 알고 가야해!

(GO)

[03강] 기억장치 관리

1. 병행프로세스
2. 교착상태
3. 주기억장치 관리전략
4. 다중연속 가변분할
5. 가상기억장치 관리기법

합격
비법



대한민국 대표 IT강사
조 대 호

여러분의 인생드라마는
지금 잘 진행되시나요?
그 드라마의 주인공은
바로 당신입니다.
이 세상 모든사람들은
당신을 위한 조연입니다.
당신의 드라마가
성공한 그날!
당신의 감사메시지 속에서
저의 이름이 불려지기를
기원합니다.

정보처리기사/산업기사 합격은 당연! 고득점 합격이 목표! (GO)

1. 병행프로세스

JODAEHO

gisa plus

❖병행 프로세스(Concurrent Process)

- 병행 프로세스란 어떤 문제를 해결하기 위해 동시에 실행되는 두 개 혹은 그 이상의 프로세스를 말한다. 실제 컴퓨터의 운영체제와 사용자 프로그램들도 여러 개의 프로세서들이 상호 협력하여 동시에 실행되고 있으며 이러한 병행 프로세스는 분산처리 시스템에서 중요한 개념이 되고 있다.

❖임계 영역(Critical section)

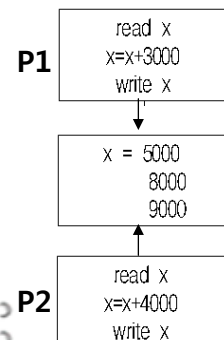
- 다중 프로그래밍 운영체제에서 한 순간에 여러 개의 프로세스에 의하여 공유되는 데이터 및 자원에 대하여, 한 순간에는 반드시 하나의 프로세스에 의해서만 자원 또는 데이터가 사용되도록 하고, 이러한 자원이 프로세스에 의하여 반납된 후, 비로소 다른 프로세스에서 자원을 이용하거나 데이터를 접근할 수 있도록 지정된 영역을 의미

❖상호 배제(Mutual exclusion)

- 공유자원을 어느 시점에서 단지 한 개의 프로세스만이 사용할 수 있도록 하며, 다른 프로세스가 공유자원에 대하여 접근하는 것을 금지하는 기법
- 세마포어(Semaphore)
 - 다익스트라에 의해 제안되었으며 반드시 상호 배제의 원리가 지켜져야 하는 공유 영역에 대하여 각각의 프로세스들이 접근하기 위하여 사용되는 두 개의 연산 P와 V라는 연산을 통해서 프로세스 사이의 동기를 유지하고 상호배제의 원리를 보장하는 것
 - 음수값을 가질 수 없다.
 - 세마포어 연산 중에는 절대 인터럽트되어서는 안 된다.
- 모니터(Monitor)
 - 순차적으로만 사용할 수 있는 공유 자원 혹은 공유 자원 그룹을 할당하는데 사용되며, 데이터 및 프로시저를 포함하는 병행성 구조
 - 모니터 내의 자원을 원하는 프로세스는 반드시 해당 모니터의 진입부를 호출함으로써 공유 자료에 접근할 수 있다.

주요포인트 꼭 알고 가야해!

병행 프로세스에서의 문제점
공용 변수(data)를 여러 개의
프로세스가 동시에 접근할
경우에 lost update problem
(갱신손실)이 발생한다.



❖ 교착상태(Deadlock)의 정의

- 둘 이상의 프로세스들이 서로 다른 프로세스가 차지하고 있는 자원을 요구하여 무한정 기다리게 함으로 인해 결국 해당 프로세스의 진행이 중단되는 현상

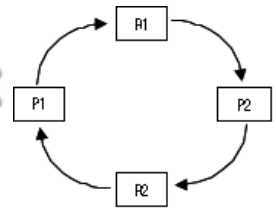
❖ 교착상태의 발생 조건

- 상호 배제(Mutual Exclusion)
 - 각각의 프로세스들이 필요한 자원에 대해 배타적 통제권을 요구할때
- 점유와 대기(Hold and Wait)
 - 프로세스가 다른 자원을 요구하면서 할당받아 점유하고 있는 자원을 해제하지 않을 때
- 비중단(Nonpreemption)
 - 프로세스에 할당된 자원은 끝날 때까지 강제로 중단할 수 없을 때
- 순환 대기(Circular Wait)
 - 프로세스 간의 자원 요구가 원형의 사슬 형태로 존재할 때(=환형 대기)

❖ 교착상태의 해결 방법

- 예방(Prevention) : 교착상태발생조건 중 하나를 부정함으로써 해결
- 회피(Avoidance) : 상태를 파악하여 발생 가능성이 있는것을 피함
- 발견(Detection) : 사이클의 유무를 판별하여 간선을 제거
- 회복(Recovery) : 강제로 종료시켜서 해결
 - 자원의 선점에 의한 회복
 - 복귀(Rollback)에 의한 회복
 - 프로세스 제거에 의한 회복

주요포인트 꼭 알고 가야해!



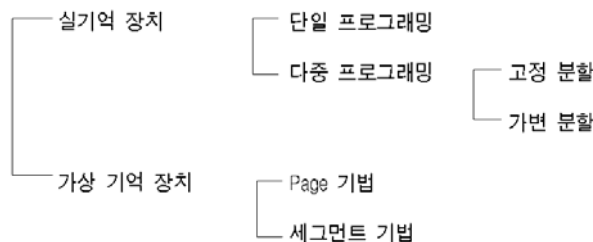
[은행원 알고리즘]

교착상태 해결방법중 회피에 해당하는 것으로 신용도가 없는 프로세스를 파악하여 피하는 방법



3. 주기억장치 관리

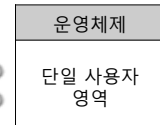
❖ 기억장치 할당



❖ 다중 연속 고정 분할(Multiple contiguous Fixed Tasks) 기법

- 기억장치를 서로 다른 고정 크기로 분할하여 다수의 프로그램이 처리되도록 이용하는 방법이다.
- 절대 번역과 로딩(분할 작업큐 방식)
 - 각각의 작업들이 절대 어셈블러와 컴파일러에 의해 번역되어 주기억장치내의 지정된 부분에서만 실행되는 기법
 - 하나의 작업이 실행준비가 되어있다 할지라도 해당되는 분할이 이미 차 있다면 그 작업은 실행될 수 없다.
 - 외부 단편화와 내부단편화가 발생한다.
- 재배치 가능 번역과 로딩(공통 작업큐 방식)
 - 재배치 컴파일러, 어셈블러와 로더에 의해 만들어진 프로그램으로서, 프로그램을 실행할 수 있을 만큼 기억공간이 존재한다면 그 부분의 위치에 상관없이 프로그램을 적재하여 실행할 수 있도록 하는 기법을 말한다.
 - 외부 단편화는 없으나 내부단편화가 발생한다.

주요포인트 꼭 알고 가야해!



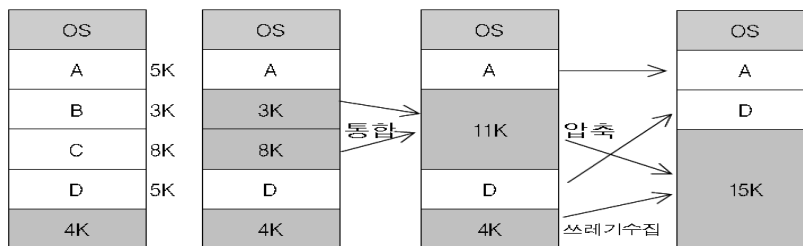
단일 분할(Single partition allocation) 기법

하나의 프로그램만이 상주하여 처리되는 형태이다. 프로그램의 크기는 기억장치보다 클 수 없지만 오버레이(Overlay)기법을 이용하여 처리가 가능하다. 운영체제와 사용자 영역의 보호는 하나의 경계 레지스터(bound Register)를 이용하면 된다.



❖ 다중 연속 가변 분할(Multiple contiguous Variable Tasks) 기법

- 다중 연속 고정 분할 프로그래밍 방법의 단점을 보완한 것으로, 미리 고정된 경계를 없애고 각 프로그램에서 필요로 하는 만큼의 기억장치를 프로그램 실행 시 할당받는 기법으로 동적 분할이라고도 한다. 내부 단편화는 없으나 외부 단편화가 발생한다.
- 외부 단편화를 해결하기 위해 아래와 같은 방법을 사용한다.
- 통합(coalescing)
 - 가변분할 다중 프로그래밍 시스템에서 하나의 작업이 끝났을 때, 그 저장 장치가 다른 비어있는 저장 장소와 인접되어 있는지를 점검한다. 이때 인접한 공백들을 하나의 공백으로 합하는 과정을 말한다.
- 압축(Compaction)
 - 메모리 관리 기법 중에서 서로 떨어져 있는 여러 개의 낭비 공간을 모아서 하나의 큰 기억 공간을 만드는 작업으로 모든 사용되고 있는 기억장소를 주기억장치의 한쪽 끝으로 옮기는 것을 말한다.
 - 일반적으로 기억장소 압축기법을 쓰레기 수집(Garbage Collection)이라고도 한다.
- 재배치(Relocation)
 - 임의의 기억장소에서 실행 중이던 프로그램을 다른 장소로 이동시키고자 할 때, 그 이동한 장소에서도 프로그램이 실행 가능하도록 하기 위해 번지(Address)에 관계된 부분을 조정해 주는 작업을 말한다.



주요포인트 꼭 알고 가야해!

5. 가상기억장치 관리기법

❖ 페이징(Paging) 기법

- 페이지(Page)라는 고정 크기 조각 단위로 나누어 할당한다.
- 주기억 공간의 적재단위를 페이지 프레임(Page-Frame)이라고 한다.
- 페이지와 프레임은 동일한 크기로 나눈다.
- 가상 주소공간내의 페이지들이 주기억장치의 어느 페이지 프레임에 사상되었는지 알기 위해 페이지 사상표(Page Mapping Table)를 이용한다.
- 외부단편화는 발생하지 않으나, 마지막 프레임에서 내부단편화는 발생할 수 있다.

❖ Page Table(Mapping Table)의 관계

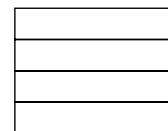
- Page의 크기가 작아지면...
 - 페이지 개수 증가, Page Table의 크기가 커짐, 유용도가 커짐, 내부 단편이 적어진다.
- Page의 크기가 커지면...
 - 페이지 개수 감소, Page Table의 크기는 작아짐, 입출력 시간이 줄어들고 내부 단편이 커진다.

❖ Segmentation

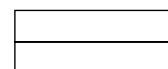
- 페이지 상호간의 결속도를 높이기 위해 사용자 주소공간을 논리적인 단위로 나눈 세그먼트를 이용하는 방식
- 세그먼트는 크기가 서로 다르고 프로그램 모듈로 표현되기도 한다.
- 기억장치 보호를 위해 기억장치 보호키를 사용한다.
- segmentation의 문제점
 - 평균 세그먼트의 크기가 매우 큰 경우 외부 단편화가 발생
 - first-fit, best-fit 알고리즘을 사용하여 어떤 세그먼트를 할당하기 위한 탐색 시간이 길어질 수 있다.

주요포인트 꼭 알고 가야해!

[보조기억장치]



[주기억장치]



[Mapping(사상)]

가상 기억장치에서 주기억 장치로 페이지를 옮겨 넣을 때 주소를 조정해 주는 것

- ① 직접사상 : Page Table을 주소(페이지번호)로 검색
- ② 연관사상 : Page Table을 내용으로 검색
- ③ 직접/연관사상 : 일부는 직접, 나머지는 연관사상 이용

주요포인트 꼭 알고 가야해!

(GO)

[04강] 기억장치 관리전략

1. 반입전략과 배치전략
2. 재배치전략1
3. 재배치전략2

합격
비법



대한민국 대표 IT강사
조 대 호

여러분의 인생드라마는
지금 잘 진행되시나요?
그 드라마의 주인공은
바로 당신입니다.
이 세상 모든사람들은
당신을 위한 조연입니다.
당신의 드라마가
성공한 그날!
당신의 감사메시지 속에서
저의 이름이 불려지기를
기원합니다.

정보처리기사/신입기사 합격은 당연! 고득점 합격이 목표! (GO)

1. 반입전략과 배치전략

JODAEHO

gisa plus

❖반입(Fetch) 전략

- 주기억장치에 넣을 프로그램이나 데이터를 언제 보조기억장치에서 주기억장치로 가져올 것인가를 결정하는 전략
- 요구 반입 전략(Demand Fetch) : 실행중인 프로세스에 의하여 호출된 자료만이 주기억장치로 옮겨진다. 가상기억장치에서 사용된다.
- 예상 반입 전략(Anticipatory) : 프로세스에 의하여 어떤 페이지나 세그먼트들이 호출될지 미리 예상하여 호출 가능성이 높은 블록을 미리 주기억장치에 옮겨 준다.

❖배치(Placement) 전략

- 인출 전략에 의해 새로 반입된 프로그램을 주기억장치의 어느 곳에 위치시킬 것인가를 결정하는 전략
- 최초 적합(First-fit) 전략 : 입력된 프로그램을 수용할 수 있는 공간 중 가장 먼저 발견된 공간에 할당하는 방법
- 최적 적합(Best-fit) 전략 : 입력된 프로그램을 수용할 수 있는 공간 중 가장 적은 공간에 할당하는 방법
- 최악 적합(Worst-fit) 전략 : 프로그램을 수용할 수 있는 공간 중 가장 큰 공간에 할당하는 방법으로 단편화 현상을 줄일 수 있음

	OS
①	8K
②	5K
③	30K

Ex> 4k의 크기를 가진 프로그램을 할당하려고 한다. 아래와 같은 방법을 이용했을 경우 어떤 위치에 할당되는가? 그리고 이때 발생하는 단편화의 종류와 크기는?

- i) 최초적합
- ii) 최적적합
- iii) 최악적합

주요포인트 꼭 알고 가야해!

[보조기억장치]

A
B
C
D

[주기억장치]

[CPU]

--

정보처리기사/신입기사 합격은 당연! 고득점 합격이 목표!

(GO)

[04강] 기억장치 관리전략

Slides 18

❖재배치(Replacement) 전략의 정의

- 새로운 페이지나 세그먼트가 적재될 주기억장치의 공간이 없을 때 주기억 장치에 있는 페이지나 세그먼트들 중에 어느 것을 제거할 것인가를 결정짓는 전략이다.

❖FIFO(First-In-First-Out) 기법

- 주기억장치에 가장 오랫동안 있었던 페이지를 교체하는 방법
- 구현하기가 쉽고 간단한 기법으로 프로세스당 더 많은 페이지 프레임이 할당될수록 더 많은 페이지 부재가 발생하는 FIFO 변칙현상이 일어난다.
→ Belady anomaly(비레디의 변위)
- 예) 참조열이 다음과 같을 때 3개의 페이지 프레임을 FIFO로 구현할 때 발생하는 Page Fault 수는?

참조열	2	3	2	1	5	2	4	5	3	2	5	2
페이지 프레임												
페이지 부재												

❖LFU(Least Frequently Used) 기법

- 각 페이지들이 얼마나 자주 사용되었는가에 중점을 두어 참조 횟수가 가장 적은 페이지를 대체시키는 기법

❖MFU(Most Frequently Used) 기법

- 각 페이지들이 얼마나 자주 사용되었는가에 중점을 두어 참조 횟수가 가장 많은 페이지를 대체시키는 기법

주요포인트 꼭 알고 가야해!

[Page Fault(페이지 부재)]
가상 기억 장치 시스템에서 가상 페이지 주소를 사용하여 데이터를 접근하는 프로그램이 실행될 때, 프로그램에서 접근하려고 하는 페이지가 주기억장치에 있지 않은 경우 발생하는 현상



❖OPT(Optional) 기법(=Belady의 MIN기법)

- 최적의 성과를 올리기 위해서 앞으로 가장 오랫동안 사용되지 않을 페이지를 대체하는 기법
- 가장 효율적이지만 구현이 불가능(참조열에 대한 미래의 지식이 필요 → 다음에 수행될 작업이 무엇인지 알 수가 없기 때문)

❖LRU(Least Recently Used) 기법

- 현 시점에서 가장 오랫동안 사용하지 않은 페이지를 교체할 페이지로 선택.
- 각 페이지의 사용시간을 기억해 두어야 하므로 Overhead가 크다.
- 예) 참조열이 다음과 같을 때 3개의 페이지 프레임을 LRU로 구현할 때 발생하는 Page Fault 수는?

참조열	2	3	2	1	5	2	4	5	3	2	5	2
페이지 프레임												
페이지 부재												

❖NUR(Not Used Recently) 기법(=Page Classes)

- 시간 오버헤드를 줄이는 기법으로서 각 페이지마다 참조 비트(reference bit)와 갱신 비트(update bit)를 두고 페이지가 참조될 때 가장 최근에 참조되지 않은 페이지를 교체하는 알고리즘이다.

0(0,0)	참조X, 변경X	2(1,0)	참조O, 변경X
1(0,1)	참조X, 변경O	3(1,1)	참조O, 변경O

주요포인트 꼭 알고 가야해!

주요포인트 꼭 알고 가야해!

[05강] 디스크 스케줄링

1. 스래싱
2. 디스크 공간할당 기법
3. 디스크 스케줄링
4. 디스크 스케줄링 예문

합격
비법대한민국 대표 IT강사
조 대 호

여러분의 인생드라마는
지금 잘 진행되시나요?
그 드라마의 주인공은
바로 당신입니다.
이 세상 모든사람들은
당신을 위한 조연입니다.
당신의 드라마가
성공한 그날!
당신의 감사메시지 속에서
저의 이름이 불려지기를
기원합니다.

정보처리기사/산업기사 합격은 당연! 고득점 합격이 목표! (GO)

1. Thrashing(스래싱)

JODAEHO

gisa plus

❖ Thrashing이란?

- 하나의 프로세스가 작업 수행 과정에서 수행하는 기억 장치 접근에서 지나치게 많은 페이지 폴트가 발생하여 프로세스 수행에 소요되는 시간보다 페이지 이동에 소요되는 시간이 더 커지는 현상
- 스래싱은 CPU의 효율을 심각하게 저하시키는 요인이 된다.

❖ 지역성(locality)

- 프로그램이 실행할 때 기억장치 내의 모든 정보를 균일하게 참조하는 것이 아니라 어느 한 순간에 특정 부문을 집중적으로 참조하는 프로그램의 순차적인 성질을 말한다.
 - ① 시간 지역성(Temporal locality) : 처음에 참조된 기억장소가 가까운 미래에도 계속 참조될 가능성이 높다는 것을 의미한다.
순환(Looping), 서브루틴(Subroutine), 스택(Stack), 계산·집계변수
 - ② 공간 지역성(Spatial locality) : 일단 어느 기억장소가 참조되면 그 근처의 기억장소가 계속 참조되는 경향이 있음을 의미한다.
배열(Array), 순차적 코드실행, 변수의 선언 부분

❖ Working set model

- 하나의 프로세스가 자주 참조하는 페이지의 집합을 의미하며, 이런 페이지 집합이 적재되면 프로세스는 한동안 페이지 폴트 없이 실행될 수 있다.
- 일반적으로 Working Set의 크기는 작을수록 좋으며, Page의 크기가 작을수록 알찬 위경셋을 구현할 수 있다.

❖ 페이지 부재 빈도

- 원하는 페이지 부재율의 상한과 하한을 정해 놓고 페이지 부재율이 상한을 넘으면 프로세스에게 다른 프레임에 더 할당해 주고 하한부 보다 낮으면 그 프로세스로부터 프레임을 회수한다. 이렇게 함으로써 스래싱을 방지하면서 페이지 부재율을 측정하고 조절할 수 있을 것이다.

주요포인트 꼭 알고 가야해!

**Prepaging(예상 페이지징
=Anticipatory paging)**

초기의 과도한 페이지 부재를 방지하기 위한 기법으로 필요할 것 같은 모든 페이지를 한꺼번에 기억장치에 갖고 오는 것으로 thrashing을 방지할 수 있고 프로세스의 실행시간을 단축시킬 수 있다.

정보처리기사/산업기사 합격은 당연! 고득점 합격이 목표!

(GO)

[05강] 디스크 스케줄링

Slides 22

❖연속 할당(Continuous allocation)

- 디스크의 연속된 공간에 할당하는 방법
- 만일 요구하는 크기의 디스크 공간이 없는 경우에는 할당할 수 없다.
- 단편화가 심하기 때문에 디스크 압축이 필요하다.
- 직접접근이 가능하기 때문에 접근시간이 빠르다.

❖불연속 할당(Linked allocation) = 연결 할당

- 링크를 이용해서 각 섹터단위로 할당하는 방법
- 한 파일의 내용은 섹터들의 링크로 연결된다.
- 디스크 압축이 필요 없다.
- 링크를 이용하기 때문에 접근시간이 느리다.

❖블록 할당의 종류

- 블록 체인 기법
- 색인 블록 체인 기법 : 색인 블록에 포인터를 모아놓고 직접 접근
- 블록 지향 파일 사상 기법 : 포인터 대신 블록번호 사용

❖파일 할당표(FAT)

- 사용자가 해당 블록의 포인터를 실수로 지워지게 하는 것을 예방하고 블록 접근을 빠르게 하기 위하여 포인터를 모아 놓은 곳
- 윈도우시스템에서 파일이나 하위 디렉토리가 디스크에서 어떤 위치에 저장되어 있는지의 위치 정보를 저장하는 테이블

주요포인트 꼭 알고 가야해!

3. 디스크 스케줄링

❖디스크 스케줄링 기법

- 다수의 사용자가 서로 다른 작업을 처리하기 위해서 디스크의 입출력을 요구할 때 좀 더 효율적으로 요청을 처리하기 위한 기법으로 처리량의 극대화, 평균응답시간의 최소화를 목적으로 한다.

❖FCFS(First Come First Served)

- 요청 대기 큐에 먼저 들어 온 요청이 먼저 서비스를 받는 기법
- 디스크 헤드가 움직이는 거리를 고려하지 않은 스케줄링 기법으로 가장 공평한 방법

❖SSTF(Shortest Seek Time First)

- 현재의 위치에서 가장 짧은 거리의 요구(탐색시간이 가장 적은 것)를 먼저 처리하는 방식으로 방향에 관계없다.
- 탐색이 한쪽으로 편중될 수 있기 때문에 헤드에서 멀리 떨어진 요청은 기아상태(starvation state)가 발생할 수 있다.

❖SCAN

- SSTF 기법의 단점인 응답시간의 편차를 해결하기 위한 방식.
- 현재 진행 중인 방향으로 가장 짧은 탐색 거리에 있는 요청을 먼저 서비스하는 기법.

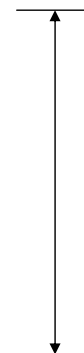
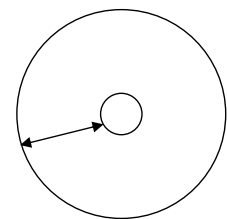
❖C-SCAN

- 헤드는 항상 바깥쪽에서 안쪽으로 이동하면서 가장 짧은 탐색 시간을 갖는 I/O 요청을 서비스하며, 더 이상 안쪽의 I/O 요청이 없을 경우에 헤드는 가장 바깥쪽의 I/O 요청으로 옮겨져서 다시 안쪽으로 I/O 요청을 서비스하는 스케줄링 기법

❖N-step SCAN

- SCAN과 동일하나 현재 대기 중인 요구만 서비스한다. 진행 중에 도착한 요구는 다음 반대 방향으로의 서비스 시에 최적으로 서비스 받을 수 있도록 재배열되는 기법이다.

주요포인트 꼭 알고 가야해!



예문> 현재 헤드의 위치가 50 에 있고 트랙 0번 방향으로 이동하며, 요청 대기 열에는 다음과 같은 순서로 들어 있다고 가정할 때, 아래의 스케줄링 알고리즘에 의한 헤드의 총 이동거리는 얼마인가?

100, 180, 40, 120, 130, 70, 80, 150, 200

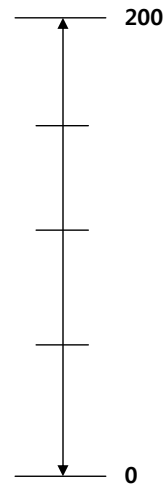
① FCFS : 50 → 100 → 180 → 40 → 120 → 130 → 70 → 80 → 150 → 200
 $50 + 80 + 140 + 80 + 10 + 60 + 10 + 70 + 50 = 550$

② SSTF : 50 → 40 → 70 → 80 → 100 → 120 → 130 → 150 → 180 → 200
 $10 + 30 + 10 + 20 + 20 + 10 + 20 + 30 + 20 = 170$

③ SCAN : 50 → 40 → 0 → 70 → 80 → 100 → 120 → 130 → 150 → 180 → 200
 $50 + 200 = 250$

④ C-SCAN : 50 → 40 → 0 → 200 → 180 → 150 → 130 → 120 → 100 → 80 → 70
 $50 + 200 + 130 = 380$

주요포인트 꼭 알고 가야해!



주요포인트 꼭 알고 가야해!

(GO)

[06강] 파일, 분산OS, UNIX

1. 파일 시스템
2. 디렉토리 시스템
3. 자원의 보호와 보안
4. 분산운영체제
5. 분산운영체제의 특성
6. 병렬처리시스템
7. UNIX
8. UNIX 명령어

합격
비법



대한민국 대표 IT강사
조 대 호

여러분의 인생드라마는
지금 잘 진행되시나요?
그 드라마의 주인공은
바로 당신입니다.
이 세상 모든사람들은
당신을 위한 조연입니다.
당신의 드라마가
성공한 그날!
당신의 감사메시지 속에서
저의 이름이 불려지기를
기원합니다.

❖파일시스템의 정의

- 운영체제에서 파일을 다루는 부분을 파일 시스템이라 하며, 파일을 안전하게 사용하고 보호될 수 있도록 무결성을 유지하는 역할을 수행한다.

❖파일 시스템의 기능

- 파일의 생성, 수정, 삭제 및 파일의 공유
- 다양한 액세스 기능과 Backup 및 복구
- 독립성을 위해 파일 명명(naming)
- 암호화와 해독기능과 사용이 편리한 인터페이스 제공

❖파일 특성

- 파일의 특성을 결정하는 기준
 - 소멸성(Volatility) : 파일에 자료를 추가하거나 파일로부터 제거하는 작업의 빈도
 - 활성율(Activity) : 프로그램이 한 번 수행되는 동안 처리되는 레코드수의 백분율
 - 크기(size) : 파일에 저장되어 있는 정보의 양
- 파일 제어블록(FCB, File descriptor)
 - 운영체제의 영역 중 현재 사용되고 있는 파일들에 대한 정보를 기록하고 있는 곳으로 시스템이 갖는 제어정보로 사용자가 참조를 못한다. 보조기억장치에 저장되어 있다가 파일이 개방(open)되면 주기억장치로 이용된다.
 - 파일명, 파일의 위치, 파일의 구조(순차, 인덱스, 직접, 분할 등)
 - 파일의 Type(데이터, object, source, execute 등)
 - 파일 작성 일자, 파일 제거 일자, 최종 수정 일자 및 시간, Access한 횟수

주요포인트 꼭 알고 가야해!

2. 디렉토리 시스템

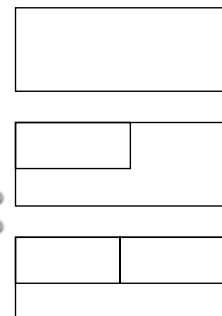
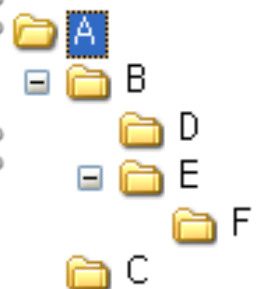
❖디렉토리의 정의

- 디스크에 수록된 파일을 찾기 위한 색인이며, 시스템이 가지고 있는 파일의 일람표이며, 파일을 관리하기 위한 테이블이다.
- 내용 : 파일 이름, 파일 형태, 저장 위치, 크기, 파일 소유자, 현재 위치, 파일 접근 정보, 작성된 시간, 변경된 시간, 파일 이용 수

❖디렉토리의 구조

- 단일 디렉토리 구조
 - 가장 간단한 디렉토리 구조로 모든 파일들이 유일한 이름을 가지고 있다.
 - 같은 디렉토리 내에 위치하여 관리되는 디렉토리 구조
- 2단계 디렉토리 구조
 - 파일 시스템에서 중앙에 마스터 파일 디렉토리가 있고, 그 아래 사용자 파일 디렉토리가 있는 구조로 다른 사용자와의 파일 공유가 대체적으로 어렵다.
 - 파일 이름이 보통 사용자이름, 파일 이름의 형태를 취하므로 파일 이름의 길이가 길어지는 디렉토리 구조
- 트리 형태의 디렉토리 구조
 - 하나의 루트 디렉토리와 여러 개의 부 디렉토리로 구성이 되고 디렉토리 탐색은 포인터로 이루어지며 UNIX 운영체제에서 사용하고 공유가 불가능하다
 - 각 디렉토리의 생성과 파괴가 쉬운 비순환(Acyclic) 그래프 디렉토리 구조이며 부 디렉토리를 공유사용이 가능하다.
 - 사이클이 없는 그래프구조를 가지고 융통성이 있으며 기억공간을 절약할 수 있으나, 복잡하고 하나의 파일에 다수의 이름이 존재할 수 있는 디렉토리 구조
- 일반적인 그래프 디렉토리 구조
 - 사이클이 허용되는 그래프 구조로 공유가 가능하며 탐색 알고리즘이 간단하며 파일로 액세스하기 쉽다.
 - 파일의 제거를 위하여 쓰레기 모음(Garbage - Collection)을 위한 참조 계수기가 필요

주요포인트 꼭 알고 가야해!



❖ 자원 보호의 정의

- 인증 받지 못한 사용자로부터 자원을 보호하기 위해 컴퓨터 시스템에 접근을 제어하는 기법
- 파일 보호 방법
 - 접근 제어(Access Control), 파일 이름 부여(Namming)
 - 암호화(Passwords), 해독화(Cryptography)

❖ 자원 보호 기법

- 컴퓨터 시스템에서 사용되는 자원들(파일, 프로세스, 메모리 등)에 대하여 불법적인 접근 방식을 목적
 - 액세스 제어 행렬(Access Control Matrix) : 열(Column)과 행(Row)의 집합으로 구성
 - 액세스 제어 리스트(Access Control List) : 각 객체에 대한 리스트는 <영역, 권한 집합>의 순서쌍으로 구성된다. 영역과 결합되어 있으나 사용자를 위해 간접적으로 액세스되는 기법을 말한다.
 - 권한 리스트 : 권한 영역에 대한 권한 리스트는 객체와 그 객체에 허용된 조작 리스트이다.

❖ 보안의 정의

- 시스템과 시스템에 있는 데이터에 결함이 없도록 신뢰성을 유지하려는 기법.

❖ 외부 보안(External security)

- 시설 보안 - 천재지변과 외부 침입자로부터의 보안
- 운용 보안 - 전산소 관리 및 경영자들의 통제에 의해서만 이루어지는 보안
- 사용자 인터페이스 보안(User interface security) - 인증 받은 사용자에게만 허가

❖ 내부 보안(Internal security)

- 프로그램의 신뢰성 있는 운영과 데이터의 무결성을 보장하기 위한 보안

주요포인트 꼭 알고 가야해!

[암호화기법(Cryptography)]

지정된 사람만이 그 데이터를 이해할 수 있도록 데이터를 변형하는 것

- ① 암호이용의 비밀 시스템
- ② 암호 분석
- ③ 공용키 시스템
- ④ 디지털 서명
- ⑤ DES(대칭 암호 기법)와 RSA(비대칭 암호 기법)

[패스워드 보호]

패스워드는 기억장치에 저장되고 컴퓨터를 사용할 때마다 암호 확인 후 사용권을 허가하는 방식이다.

[고장 허용 시스템(Fault-Tolerant System)]

시스템의 부품 중에 일부가 고장이 나더라도 시스템이 정상적으로 작동할 수 있는 시스템을 말한다.

4. 분산운영체제

❖ 강결합 시스템(Tightly Coupled)

- 모든 프로세서와 시스템 하드웨어의 관리가 동일 운영체제 하에서 하나의 메모리를 공유하는 시스템으로 다중처리 시스템이라고도 한다.

❖ 약결합 시스템(Loosely Coupled)

- 둘 이상의 독립된 컴퓨터 시스템을 통신 링크를 통하여 연결한 시스템으로 분산 처리 시스템이라고도 한다.

❖ 시분할 또는 공유버스

- 다수의 프로세서, 기억장치 및 입출력 프로세서들이 공통의 버스를 통해 연결된 것으로, 각 장치간의 자료 전송은 장치 자체의 버스 인터페이스에 의하여 제어.
- 하나의 공통 버스를 모든 프로세서가 공유하고 있으므로 접근 충돌이 발생
- 버스에 장치들을 직접 부착시킴으로 쉽게 추가할 수 있어 경제적이고 단순하며 융통성.

❖ 크로스바 교환 행렬

- 크로스바 구조는 프로세서 버스와 기억장치 모듈 통로간의 교차점에 위치한 다수의 교차점(cross point)으로 구성되며 공유 버스 시스템에서 메모리 수만큼 버스의 수를 늘리고 여러 개의 기억장치와 프로세서를 연결한다.
- 경제적이고 간단하며 융통성이 있다.

❖ 다중 출입구 기억장치

- 크로스바 교환 행렬의 복잡한 스위치 구조를 개선한 형태로서 여러 개의 공유버스를 통하여 모든 장치들이 연결된다.

주요포인트 꼭 알고 가야해!

❖분산 운영체제의 설계 쟁점

- 분산 운영체제는 중앙 집중식에 비하여 소프트웨어의 개발이 어렵다.

❖분산운영체제의 특성

- 신뢰성, 융통성, 성능, 확장성
- 투명성
 - 사용자는 분산을 느끼지 못하고 단일 처리기 시스템에서 작동하는 운영체제의 형태처럼 느끼게 하는 것.
- ① 위치 투명성 : 사용자는 원하는 자원의 위치를 알 필요가 없다.
- ② 접근 투명성 : 프로세스는 지역과 전역 모두에 대하여 같은 형식의 접근 방법을 가진다.
- ③ 이주 투명성 : 사용자나 응용 프로그램의 동작에 영향을 받지 않고 시스템 내에 있는 정보 객체를 이동할 수 있도록 하는 투명성(Transparency)이다.
- ④ 복사투명성 : 사용자에게 통지할 필요 없이 시스템 안에 파일들과 자원들의 부가적인 복사를 자유로 할 수 있다.
- ⑤ 병행투명성 : 다중 사용자들이 자원들을 자동으로 공유할 수 있다.
- ⑥ 중첩투명성 : 여러 개의 정보 객체의 인스턴스가 사용자나 응용 프로그램에 의해 중첩된 사실을 모르더라도, 신뢰성과 성능을 향상시키기 위하여 사용.

❖분산 운영체제의 이주(Migration)

- ① 자료 이주 : 자료를 원하는 지역으로 전송하는 방법으로 실제로 필요한 파일의 일부만을 전송
- ② 연산 이주 : 파일이 존재하는 지역으로 접근하여 처리를 한 후 그 결과값만을 다시 원래의 지역으로 반환하는 방법(처리량이 많을 때 사용)
- ③ 프로세스 이주 : 원하는 작업을 다른 지역으로 이동시켜 실행하는 것이 유리한 경우에 사용하는 방식(부하균등, 연산 속도 향상)

주요포인트 꼭 알고 가야해!

6. 병렬처리 시스템

❖병렬처리 시스템

- 컴퓨터 시스템의 계산 속도를 향상시키기 위해서 동시에 데이터를 처리할 수 있는 기능을 제공하는 기술
- 여러 개의 프로세서가 동시에 정보를 처리하는 컴퓨터 시스템
- 목적 : 처리속도 향상과 처리량(Throughput) 증가

❖주종(Master/Slave) 다중 처리기 시스템

- 가장 구현하기 쉬운 다중처리 시스템의 한 형태로 흔히 기존의 다중 프로그래밍 시스템의 직접적인 확장으로 구현.
- 하나의 주프로세서(Master)와 여러 개의 종프로세서(Slave)로 구성된다.
- 주프로세서는 입출력과 연산을 담당하고 종프로세서는 연산만을 담당한다.

❖분리 수행(Separate execution)

- 각 프로세서가 독립적으로 단일 시스템처럼 독자적인 운영체제와 기능을 가지는 형태이다.
- 주종보다 신뢰도가 높아서 한 프로세서의 고장이 전체 시스템에 영향을 주지 못한다.

❖대칭적 처리(symmetric treatment)

- 모든 프로세서가 동등한 입장의 대칭성을 가지는 복잡한 형태의 시스템
- 한 운영체제를 동시에 수행할 수 있기 때문에 상호배제가 필요하다.
- 보다 정확하게 작업부하를 분산시킬 수 있으며 가장 신뢰성이 높으며 자원의 활용도가 높다.
- 한 순간에 하나의 프로세서만이 관리 프로세서가 되기 때문에 정보를 일관성 있게 운영할 수 있다.

주요포인트 꼭 알고 가야해!

❖UNIX의 특징

- 대화식(Interactive)으로 동작하는 시분할 시스템(time sharing system)
- 다중 작업(Multi-Tasking), 다중 사용자(Multi-User) 환경
- C언어로 작성되어 사용과 수정이 쉽고 확장성(이식성)이 좋다.
- 파일의 구조는 계층적 트리(tree) 구조로 되어 있다.

❖UNIX 커널(Kernel)

- 항상 주기억장치에 상주하는 운영체제에서 가장 핵심적인 부분으로 컴퓨터 시스템, 즉 하드웨어를 관리한다.
- 시스템의 하드웨어를 제어하는 임무로 메모리, CPU, 단말기, 프린터 등 시스템의 자원 활용도를 높이기 위해 스케줄링과 자료관리를 하는 핵심요소
- 대부분 C언어로 개발되었기 때문에 이식성과 확장성이 뛰어나.
- 기능 - 프로세스간 통신(IPC), 프로세스 관리(process management), 기억장치 관리(memory management), 입출력 관리, 타이머 관리

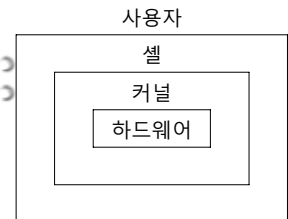
❖UNIX 셸(Shell)

- 사용자와 UNIX 사이의 대화를 위한 명령어 체계
- 사용자 명령의 입력을 받아 시스템 기능을 수행하는 명령 해석기로서 사용자와 시스템 간의 인터페이스를 담당하는 것

❖i-node

- UNIX에서 각 파일에 대한 정보를 기억하고 있는 자료구조로서 파일 소유자의 식별번호, 파일 크기, 파일의 최종 수정시간, 파일 링크 수 등의 내용을 가지고 있는 것
- i-node에 포함된 정보
 - 파일이 가장 최근에 사용된 시간, 파일이 변경된 가장 최근의 시간, 파일 소유자의 사용자 번호, 파일이 만들어진 시간, 파일의 링크수, 크기, 파일의 내용이 담긴 디스크상의 실제 주소, 파일 소유자의 사용자 식별, 보호 비트

주요포인트 꼭 알고 가야해!



8. UNIX 명령어

❖시스템 호출(System Call) 명령

- Exec : 프로세스가 지정된 파일을 실행할 수 있게 한다.
- Fork : 새로운 프로세스를 생성(복제)하는 명령
- Finger : 각 개인의 정보를 알아내기 위한 명령
- Pipe : 프로세스 간 통신을 위한 시스템 호출문, 한 프로세스의 출력이 다른 프로세스의 입력으로 사용

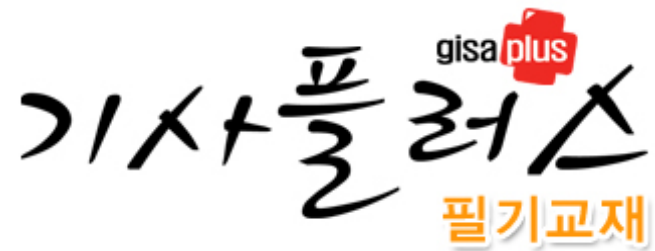
❖사용자를 위한 일반 명령

- at : 지정된 시간에 명령어를 실행
- cat : file의 내용을 화면에 표시
- cd : 디렉토리 변경
- cp : 파일의 복사
- kill : 프로세스를 끝냄
- Grep : 파일 내에서 사용자가 찾고자 하는 특정 패턴이나 문자열을 검색한다.
- Pwd : 현재 작업중인 디렉토리 경로를 보여준다.

❖시스템 관리 명령

- Chmod : 파일이나 디렉토리에 대한 접근 허가 모드를 변경
- Mount : 기존 파일 시스템에서 새로운 파일 시스템을 서브디렉토리로 연결할 때 사용
- Su : 슈퍼 사용자 또는 다른 사용자 모드로 변경한다.
- Devinfo : 시스템에 설치된 디스크 장치에 관한 정보를 출력한다.
- Killall : 활동중인 모든 프로세스를 강제로 종료시킨다.

주요포인트 꼭 알고 가야해!



정보처리기사/산업기사 **필기**

데 이 터 통신

정 보 통 신 개 론



주요포인트 꼭 알고 가야해!



[01강] 통신의 개념

1. 통신과 정보통신
2. 정보통신의 형성과 발전
3. 정보통신 시스템의 이용
4. 정보통신 시스템의 구조
5. 단말기
6. 접속장비(인터넷워킹)



대한민국 대표 IT강사
조 대 호

여러분의 인생드라마는
지금 잘 진행되시나요?
그 드라마의 주인공은
바로 당신입니다.
이 세상 모든사람들은
당신을 위한 조연이랍니다.
당신의 드라마가
성공한 그날!
당신의 감사메시지 속에서
저의 이름이 불려지기를
기원합니다.

데이터통신 합격은 당연! 고득점 합격이 목표!



1. 통신과 정보통신

JODAEHO

gisa plus

❖데이터와 정보의 변천

- 데이터(Data) → 정보(Information) → 지식(Knowledge) → 지능(Intelligence)

❖통신의 개념

- 사람과 사람 사이의 의사소통이나 정보를 교환하는 것
- 한 지점에서 다른 지점까지 정보를 보다 투명하고 빠르게 전송하는 것

❖정보통신의 개념

- 데이터 전송과 데이터 처리를 유기적으로 결합하도록 시스템을 구성하여 정보 전달의 목적을 달성
- 다량의 정보를 신속하게 전송
- 컴퓨터 자원의 공유 및 비용절감
- 거리와 시간의 극복

❖정보통신 = 정보 전송 기술 + 정보 처리 기술

❖정보통신 서비스의 형태

- 5가지 형태의 정보통신, 즉 음성통신, 데이터통신, 화상(이미지)통신, 영상통신, 멀티미디어 통신을 총칭
- 정확성, 효율성, 보안성을 목적으로 한다.

❖정보통신의 구성

- 정보원, 전송매체 정보목적지

주요포인트 꼭 알고 가야해!

[정보와 데이터]

정보 : 데이터를 가공처리하여 얻은 지식
데이터 : 가공되기 전의 숫자, 문자, 부호로된 것



데이터통신 합격은 당연! 고득점 합격이 목표!



[01강] 통신의 개념

JODAEHO

❖정보통신의 역사

- 1세대 : 부호를 이용하여 정보를 전달하는 세대
- 2세대 : 음성을 이용하여 정보를 전달하는 시대
- 3세대 : 전자 신호 및 무선을 이용하여 정보를 전달하는 세대
- SAGE : 미공군 군사용 방공시스템, 세계최초의 데이터 통신 시스템
- SABRE : 세계최초의 상업용 데이터 통신 시스템
- ARPANET : 미국방성의 세계최초의 패킷교환망(인터넷의 효시)
- ALOHA : 최초의 라디오 패킷 통신망

❖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

❖정보통신시스템의 특징

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

주요포인트 꼭 알고 가야해!

데이터통신 합격은 당연! 고득점 합격이 목표!



[01강] 통신의 개념

JODAEHO

❖정보통신 시스템의 처리 형태

- 원격 일괄 처리 시스템(Batch Processing System)
 - 자료를 1주일 또는 1개월 등의 기간 단위로 모아 두었다가 처리하는 방식
- 실시간 처리 시스템(Real Time Processing System)
 - 데이터가 발생할 때마다 처리, 은행 On-line 예금처럼 즉시 그 결과를 나타내도록 하는 방식
- 시분할 처리 시스템(Time Sharing System)
 - 한 시스템을 여러 명의 사용자가 공유하여 동시에 작업을 수행하는 방식

❖정보통신 시스템의 이용형태

- 데이터 수집 : 원격 검침, 판매 시점 관리 등
- 조회 응답 : 컴퓨터 시스템의 자료를 조사하여 해당 정보를 제공하는 정보 검색 시스템
- 메시지 교환 : 회선교환, 축적 교환
- 거래처리 : 은행 창구 업무, 좌석 예약 실무 등
- 원격 작업 입력 : 원격지로부터의 작업 발생을 위한 프로그램과 데이터 동시 수행
- 컴퓨터 네트워크 : 분산 처리 기능, 분산 데이터베이스 기능

주요포인트 꼭 알고 가야해!

데이터통신 합격은 당연! 고득점 합격이 목표!



[01강] 통신의 개념

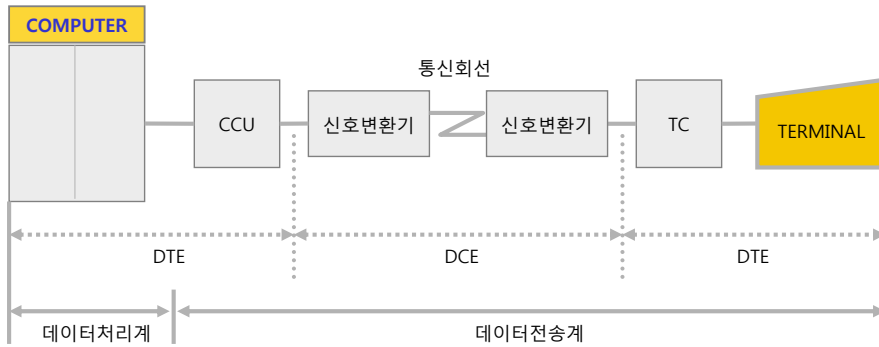
JODAEHO

❖정보통신 시스템의 유형

- 데이터 전송계 : 정보의 이동을 담당
 - 단말 장치, 신호 변환 장치(변복조기(MODEM) 또는 DSU), 통신 회선, 통신 제어 장치(CCU)
- 데이터 처리계 : 정보의 가공, 처리, 보관 등의 기능 수행, 통신처리, 정보처리
 - 중앙 처리 장치(CPU), 주변 장치

❖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)



주요포인트 꼭 알고 가야해!

[데이터 링크 제어가 통신 채널을 관리하는 단계]

회선연결
↓
링크확립
↓
정보전송
↓
링크절단
↓
회선절단



데이터통신 합격은 당연! 고득점 합격이 목표!



[01강] 통신의 개념

JODAEHO

5. 단말기(Terminal)

❖단말기기(Terminal)의 정의

- 시스템 또는 통신망의 한 지점에서 데이터가 그 시스템 또는 통신망에 들어가거나 그것으로부터 나오는 장소
- 통신로를 통하여 정보를 송신 또는 수신할 수 있는 장치
- 수행될 작업과 관련된 환경에서 데이터를 입출력 할 수 있는 장치. 입력을 전송할 수도 있고 출력을 받을 수도 있다

❖범용 단말장치

- 표시장치 : 음극선관(CRT), 액정디스플레이(LCD)
- 인식 장치 : MICR(자기잉크 문자 판독기), OCR(광학 문자 판독기), OMR(광학 마크 판독기)
- 인쇄장치 : 과거에는 충격식 주로 사용, 현재는 비충격식 주로 사용
- 자기 기억 장치 : 자기테이프, 자기디스크 장치

❖전용 단말장치

- 은행단말기, 의료/정보용 단말기, 증권/주가용 단말기, 생산 관리용 단말기,
- POS 단말기, 교통용 단말기, 교육용 단말기

❖복합 단말장치

- 지능 단말기 : 프로그램이 내장되어 있어 데이터의 입출력 및 간단한 데이터 처리능력을 가진 단말장치, 호스트와 연결되어 있지 않더라도 사용가능
- 비지능형 단말기 : 기억기능없이 단순한 입출력만 가능한 단말기(더미(Dummy))
- 원격 일괄(Remote Bath) 처리 단말기 : 주로 원격지의 일괄처리에 사용되는 장치로 고속 회선으로 연결

주요포인트 꼭 알고 가야해!

데이터통신 합격은 당연! 고득점 합격이 목표!



[01강] 통신의 개념

JODAEHO

❖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
- 패킷교환망에서 적절한 전송로를 찾아가는 것을 라우팅이라 한다.

❖리피터(Repeater)

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

❖게이트웨이(Gateway)

- 다른 네트워크로 들어가는 입구 역할을 하며 서로 다른 프로토콜을 연결할때 사용.

❖스위치(Switch)

- 네트워크 트래픽의 흐름을 통제하기 위해 각 패킷에 있는 정보를 사용하는 방식

❖허브(Hub)

- LAN에서 수신한 모든 정보를 모든 포트에 전달하며 네트워크의 확장이 용이하지만 불필요한 트래픽이 발생한다.

주요포인트 꼭 알고 가야해!

[트래픽]

어떤 통신장치나 시스템에 걸리는 부하(負荷)를 말한다. 네트워크를 통해 움직이는 데이터의 양이나 어떤 종류의 트랜잭션 및 메시지 등의 량을 나타낼 때에도 사용된다

[트래픽제어기술]

Flow control(흐름제어)
혼잡제어
교착상태 회피



데이터통신 합격은 당연! 고득점 합격이 목표!



[01강] 통신의 개념

JODAEHO

주요포인트 꼭 알고 가야해!



[02강] 정보의 전송

1. 정보 전송의 개요
2. 정보전송기기
3. 다중화기
4. FDM과 TDM
5. 집중화기
6. 전송매체

합격
비법대한민국 대표 IT강사
조 대 호

여러분의 인생드라마는 지금 잘 진행되시나요?
그 드라마의 주인공은 바로 당신입니다.
이 세상 모든사람들은 당신을 위한 조연이랍니다.
당신의 드라마가 성공한 그날!
당신의 감사메시지 속에서 저의 이름이 불려지기를 기원합니다.

데이터통신 합격은 당연! 고득점 합격이 목표!



❖정보의 전송

- 정보를 원격지까지 전송하기 위해서 모든 정보는 정보통신 장비가 다룰 수 있는 전기 신호 형태로 바꾸어 전달됨

❖신호의 특성

- 진폭 : 신호의 높이, 파동의 최고점에서 최저점까지의 거리, 소리의 크기, 단위(dB)
- 주파수 : 1초 동안에 반복되는 횟수, 단위(Hz, KHz, GHz)
- 위상 : 1주기 내에서 시간에 대한 상대적 편차

❖아날로그 신호와 디지털 신호

- 아날로그 신호
 - 사람의 목소리와 같은 음향 정보는 소리의 고저와 음폭에 의해 서로 다른 소리로 구별됨
 - 가청주파수 : 20~20000Hz
 - 음성주파수 : 300~3400Hz
 - 대역폭 : 어떤 주파수의 범위, 즉 스펙트럼의 폭이며 상한 주파수와 하한 주파수의 값의 차이로 구함
- 디지털 신호
 - 데이터 정보와 이미지 정보는 일반적으로 0과 1로 구성된 디지털 신호로 표현
 - 장거리 전송을 위해 중계기(Repeater) 필요

주요포인트 꼭 알고 가야해!

[전파대역]

HF : 3~30 Mhz
 VHF : 30~300 Mhz
 UHF : 300~3000 Mhz
 SHF : 3000~30000 Mhz

[단위]

킬로<메가<기가<테라



데이터통신 합격은 당연! 고득점 합격이 목표!



[02강] 정보의 전송

JODAEHO

2. 전송기기

❖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
- 기능 : 변조(디지털→아날로그), 복조(아날로그→디지털):원래의 파형으로 복원
- 종류
 - 모뎀(Modulation DEModulation,변복조기) : 컴퓨터의 디지털 신호를 전송회선의 아날로그로, 아날로그신호를 디지털로 바꾸어주는 변환장치
 - 디지털 서비스 유닛(DSU:Digital Service Unit) : 디지털 데이터를 원래 전송에 적합한 디지털 신호로 변환하여 디지털 전송로를 통해 전송
 - 음향결합기(Acoustic Coupler) : 단말장치와 전화기를 연결하기 위한 모뎀의 일종으로 전화기의 송수화기를 음향결합기에 결합시켜 디지털 신호를 아날로그 신호로 변환하여 전송
 - 코덱(CODEC) : 아날로그 신호를 디지털회선으로 전송하기 위해 사용

❖커넥션의 신호선

- DTR : DTE가 정상적인 동작상태에 있음을 DCE에게 통보
- DSR : DCE가 송신할 준비의 완료여부를 DTE에게 통보
- DCD : DCE가 선로쪽으로부터 감지할 수 있는 크기의 신호를 수신하고 있음을 DTE에게 통보
- TXT : 데이터를 전송할 목적으로 DTE에서 DCE로 이송

주요포인트 꼭 알고 가야해!

[25핀 컨넥터]



[RS-232C 25핀의 기능]

2번 : 송신 데이터(TXD)
 3번 : 수신 데이터(RXD)
 4번 : 송신요구(Request to send)
 5번 : 송신종료(Clear to send)

데이터통신 합격은 당연! 고득점 합격이 목표!



[02강] 정보의 전송

JODAEHO

❖ 다중화의 의미

- 여러 개의 단말기 등이 하나의 통신회선으로 데이터를 전송하는 장치

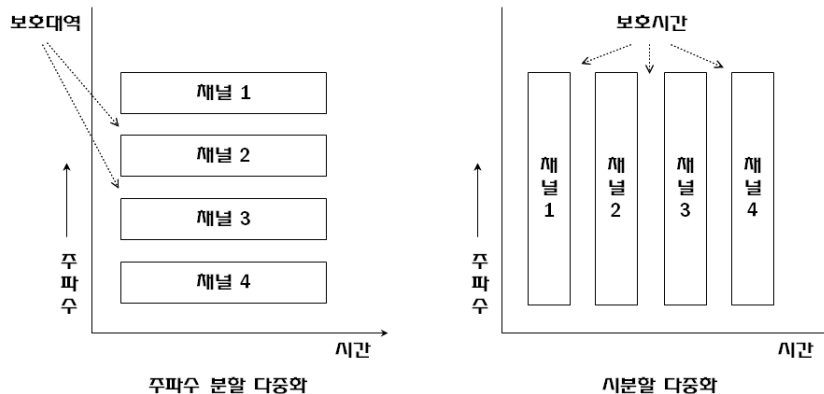
❖ 다중화기의 종류

- 주파수 분할 다중화기(FDM)
 - 하나의 물리적 통신 채널을 여러 주파수 채널로 나누어 사용
 - 채널간의 완충 지역으로 가드 밴드(Guard band)가 필요함
-1200baud 이하의 비동기예만 사용되며, 구조가 간단함
- 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
 - 동기식 시분할 다중화
 - 타임슬롯을 모든 이용자에게 규칙적으로 할당
 - 단점 - 대역폭이 낭비
 - 비동기식 시분할 다중화
 - 전송할 데이터를 갖고 있는 사용자에게만 타임슬롯을 할당하고 여유 있는 슬롯은 다른 이용자에게 할당
 - 대역폭의 이용 효율을 높임, 통계적 시분할 다중화 방식

주요포인트 꼭 알고 가야해!

4. FDM과 TDM의 비교

❖ FDM과 TDM의 비교



주요포인트 꼭 알고 가야해!

❖ 그 외 다중화기

- 역 다중화기(Demultiplexer)
 - 두개의 음성 대역폭을 이용하여 광대역의 통신 속도를 이용하는 다중화 방식
- 지능 다중화기(Intelligent Multiplexer)
 - 비동기식 시분할 다중화, 통계적 시분할 다중화라고도 한다.
 - 실제보낼 데이터가 있는 회선에만 동적으로 전송하여 효율적인 전송이 가능
- 광대역 다중화기(Group Band Multiplexer)
 - 여러가지 동기식 데이터를 묶어서 광대역으로 전송하는 방식

❖ 집중화기

- 하나의 고속 통신 회선(전송 회선)에 다수의 저속 통신 회선을 접속하기 위한 장치로 통신량이 많을 경우에 적합함(통신 선로를 공동으로 이용하기 위한 장비)

❖ 종류

- 전단의 처리기, 선로공동이용기, 모뎀공동이용기, 포트공동이용기

❖ 다중화기와 집중화기의 비교

	다중화기	집중화기
목적	데이터 전송의 효율성	
속도	저속과 고속 단말기 채널의 속도 합이 같음	저속 단말기의 속도 합이 고속 채널의 속도 합보다 크거나 같음
대역폭	입력과 출력의 대역폭이 같음	입력과 출력의 대역폭 다름
전송	규칙적 데이터 전송	불규칙적 데이터 전송

주요포인트 꼭 알고 가야해!

데이터통신 합격은 당연! 고득점 합격이 목표!



[02강] 정보의 전송

JODAEHO

6. 전송매체

❖ 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
- 광섬유 케이블(Optical Fiber Cable)
 - 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)

❖ 비유도(무선)매체

- 전송하는 안테나에 의해 발생하는 신호의 스펙트럼 혹은 주파수 대역폭이 전송 특성을 결정
- 방향성을 가지고 주파수가 낮을수록 다방향성을 띠고 주파수가 높을수록 신호를 한 방향을 지닌 방향성 전파로 전송
- 지상 마이크로파, 위성 마이크로파, 라디오

주요포인트 꼭 알고 가야해!

[광섬유의 손실]
 접속손실
 마이크로벤딩손실
 흡수손실
 레일리산란손실
 고유손실
 레일리분산손실(X)



데이터통신 합격은 당연! 고득점 합격이 목표!



[02강] 정보의 전송

JODAEHO

주요포인트 꼭 알고 가야해!

(GO)

[03강] 정보 전송 기술

1. 정보의 전송방식
2. 정보의 변조방식
3. 정보의 전송형태
4. 통신방식
5. 전송속도
6. 채널용량

합격
비법



대한민국 대표 IT강사
조 대 호

여러분의 인생드라마는
지금 잘 진행되시나요?
그 드라마의 주인공은
바로 당신입니다.
이 세상 모든사람들은
당신을 위한 조연이랍니다.
당신의 드라마가
성공한 그날!
당신의 감사메시지 속에서
저의 이름이 불려지기를
기원합니다.

데이터통신 합격은 당연! 고득점 합격이 목표!

(GO)

1. 정보의 전송 방식

JODAEHO

gisa plus

❖ 폴링과 셀렉션

- 폴링 : 주국이 데이터를 보내기 위한 시스템으로 종국이 주국으로 송신할 데이터가 있는지를 확인하는 방식
- 셀렉션 : 주국이 종국에게 데이터를 전송하기위한 방식으로 종국이 데이터를 수신할 준비가 되어있는지를 확인하는 방식

❖ 아날로그 전송

- 아날로그나 디지털 형태의 정보들은 전송 매체를 통해 전달될 때 전자파의 형태, 즉 신호 형태로 전송

❖ 디지털 전송

- 음성 정보와 같은 아날로그 정보를 디지털 신호로 변환하여 전송하는 형태를 말함

❖ 직렬/병렬 전송방식

- 직렬(Serial) 전송 : 동일한 전송선을 통해서 한 비트씩 전송하며 원거리 전송에 적합
- 병렬(Parallel) 전송 : 비트 블록 각각에 대응되는 전송선이 있어 한 번에 전송하며 근거리 전송에 적합 주로 컴퓨터와 프린터 장치의 전송 방식에 이용

❖ 동기/비동기 전송 방식

- 비동기(Asynchronous) 전송
 - 전송할 데이터를 한 번에 한 글자씩 한 문자 단위로 전송하는 방식
 - 송수신측의 동기화를 위해 시작비트(Start Bit)와 정지 비트(Stop Bit) 덧붙여 전송
- 동기(Synchronous) 전송
 - 전송할 데이터를 여러 블록으로 나누어서 블록단위로 전송하는 방식
 - 전송효율이 좋고 고속 전송에 적합

주요포인트 꼭 알고 가야해!

신호가 아날로그 형태로 전송되는 것을 '아날로그 신호 전송', 디지털 형태로 전송되는 것을 '디지털 신호 전송'이라 합니다.



[비동기(Asynchronous)]

1,200bps 이하의 저속인 FSK(주파수 편이 변조) 변조 방식을 사용

[동기(Synchronous)]

2,400bps 이상의 고속 PSK(위상 편이 변조) 변조 방식을 사용

데이터통신 합격은 당연! 고득점 합격이 목표!

(GO)

[03강] 정보전송기술

JODAEHO

❖디지털 데이터의 아날로그 부호화(MODEM 이용)

- 진폭편이변조(ASK : Amplitude Shift Keying) : 0과 1을 반송파의 진폭만 변화시켜 표현하는 방법
- 주파수편이변조(FSK : Frequency Shift Keying) : 0과 1을 서로 다른 주파수를 갖는 반송파로 변화시켜 표현하는 방법
- 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

❖디지털데이터의 디지털 부호화(DSU에 이용)

- NRZ(NonReturn-to-Zero) : 송수신전압이 0V인 경우 이를 데이터로 인정하지 않음
- RZ(Return to Zero) : 송수신 전압이 0V인 경우 이를 데이터로 인정하는 방식
- Bipolar : 신호가 양극과 음극을 모두 갖고 있는 방식
- Unipolar : 신호가 양극과 음극중 어느 한 극성만 갖고 있는 신호

주요포인트 꼭 알고 가야해!

[동기식과 비동기식]

동기식 : 동기문자를 이용하여 송,수신측의 동기를 유지하는 방식

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



❖아날로그 데이터의 디지털 부호화(펄스 변조 방식)

- PCM(Pulse Code Modulation) 방식
 - 아날로그 정보를 디지털 정보인 펄스 부호로 변환하여 전송하고 수신측에서는 디지털 정보를 원래의 파형인 아날로그 정보로 변조시키는 방식
 - 표본화(Sampling) : 샘플을 추출하여 샘플마다 값을 부여
 - 양자화(Quantization) : 샘플값을 평준화하여 표본화된 값을 추출
 - 부호화(Coding) : 양자화된 값을 디지털 신호로 변환
 - 복호화(Decoding) : 전송된 디지털신호를 원래의 펄스신호로 복원
- 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

❖베이스밴드 전송방식

- 컴퓨터나 단말장치의 디지털 출력 신호를 변조하지 않고 전송로를 통하여 그대로 전송하는 방식

❖브로드밴드 전송방식

- 주파수 분할 다중화 기법을 이용해 하나의 전송매체에 여러 개의 채널을 제공하는 것을 말한다.
- 데이터통신에서 디지털 데이터를 모뎀을 이용하여 아날로그 데이터로 변조하여 전송하며 베이스밴드보다 더 빠르고 신뢰성이 높다.

주요포인트 꼭 알고 가야해!

[양자화 잡음]

양자화 과정에서 반올림하여 발생하는 오차

[표본화 정리]

원 신호를 2배의 빈도로 표본화 하면 원래의 신호는 완전히 복원된다.

[기출문제]

PCM 전송방식에서 신호의 최대주파수가 1000[Hz]일 때 표본화 주기[us]로 적합한 것은?

정답 : 500



❖ 단방향 통신(Simplex)

- 송신자와 수신자가 고정되어 있으며, 항상 한 방향으로만 데이터를 전송하고 2선식으로 구성되어 있음(TV, 라디오 방송 등)

❖ 양방향 통신(Duplex)

- 반이중 방식(Half Duplex)
 - 양쪽의 교신자가 양방향으로 전송하나, 어느 순간에는 한 방향으로만 전송이 일어나 동시에 양방향으로 전송하는 것은 불가능(무전기) - 2선식
- 전이중 방식(Full Duplex)
 - 통신을 하는 각각의 상대방이 독립적으로 동시에 양방향으로 데이터 전송을 수행 하도록 구성(전화기 등) - 4선식

❖ 회선 접속 방식

- 포인트 투 포인트
 - 중앙의 컴퓨터와 각 단말기 등이 독립적인 회선
 - 일 대 일로 연결된 형태의 회선
- 멀티포인트
 - 1개의 회선에 여러 대의 단말기가 연결된 형태의 회선
 - 멀티드롭 회선이라고도 함

❖ 회선의 형태

- 교환회선 : 교환기를 경유하는 방식으로 전화망을 이용하며 회선교환, 메시지교환, 패킷교환방식으로 나눈다.
- 전용회선 : 교환기를 경유하지 않고 데이터 양이 많을 경우에 효율적이다.

주요포인트 꼭 알고 가야해!

[PBX]
사설 구내 교환 시스템

데이터통신 합격은 당연! 고득점 합격이 목표!



[03강] 정보전송기술

JODAEHO

❖ 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.

❖ Bps 와 baud와의 관계표

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

❖ 베어러속도

- 베이스밴드 전송방식에서 데이터 신호외의 동기신호, 상태신호 등을 포함한 전송속도로 단위는 bps를 사용한다.

주요포인트 꼭 알고 가야해!

[2진폭 위상]
2진폭 4위상 : 8위상
2진폭 8위상 : 16위상

데이터통신 합격은 당연! 고득점 합격이 목표!



[03강] 정보전송기술

JODAEHO

❖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)
- 대역폭과 신호 세력이 높을수록, 잡음 세력이 낮을수록 채널 용량은 좋음

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

- C : 통신로 용량, W : 대역폭, S : 신호세기, N : 잡음

❖대역폭 증가 기술(대역 확산 기술)

- 송신상의 보안을 유지하고 잡음이나 전송로 특성의 변화에 영향을 덜 받고 확실한 통신을 위하여 전송하는데 필요한 최저 대역보다 훨씬 넓은 주파수대역을 사용하여 확산 시킨 신호를 사용하는 기술

❖다중접속방식

- FDMA(주파수분할 다중접속방식)
- TDMA(시간분할 다중접속방식)
- CDMA(코드분할 다중접속방식) - 이동전화 시스템의 상용화 기술

주요포인트 꼭 알고 가야해!

[통신로 용량을 증가시키는 방법]

주파수 대역폭을 높인다.
신호세력을 높인다
잡음 세력을 낮춘다
신호 대 잡음의 비를 높인다.



데이터통신 합격은 당연! 고득점 합격이 목표!



[03강] 정보전송기술

JODAEHO

주요포인트 꼭 알고 가야해!



[04강] 에러제어와 프로토콜

1. 전송장애
2. 에러검출기법
3. 에러제어방식
4. 통신 프로토콜의 개념
5. 프로토콜의 전송방식
6. HDLC프레임의 구성

합격
비법대한민국 대표 IT강사
조 대 호

여러분의 인생드라마는
지금 잘 진행되시나요?
그 드라마의 주인공은
바로 당신입니다.
이 세상 모든사람들은
당신을 위한 조연입니다.
당신의 드라마가
성공한 그날!
당신의 감사메시지 속에서
저의 이름이 불려지기를
기원합니다.

데이터통신 합격은 당연! 고득점 합격이 목표!



❖ 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
- 지연왜곡 : 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)
- 충격성 잡음(Impulse Noise) : 선로의 파괴나 손상에 의해 발생
- 상호 변조 잡음 : 동일 전송 매체를 공유하는 서로 다른 주파수를 갖는 신호 사이에 발생

❖ 누화(Cross Talk)

- 인접 선로의 상호간섭에 의한 발생 즉 일종의 혼선을 말한다.

주요포인트 꼭 알고 가야해!

데이터통신 합격은 당연! 고득점 합격이 목표!



[04강] 에러제어와 프로토콜

JODAEHO

2. 에러검출기법

❖ 패리티 검사(Parity Check)

- 한 블록의 데이터 끝에 패리티 비트(Parity Bit)를 추가하는 것으로, 가장 간단한 에러 검출 기법
- 오류검출은 가능하지만 정정은 할수 없다.
- 짝수개의 비트가 동시에 오류가 발생하면 검출자체가 불가능하다.
- 1의 개수에 따른 분류로 기수(odd)검사와 우수(even)검사가 있다.

❖ 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

❖ 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
- 블록합검사의 단점을 해결하며 HDLC의 FCS에서 사용된다.
- 여러 개의 에러 검출과 수정이 가능

❖ 해밍코드(Hamming Code)방식

- 오류검출 및 교정을 할수 있다.
- 7개의 비트로 구성되며 8421로 코드화하여 3/8/6/7번째 비트열에 배열하고, 1/2/4번째 비트에 패리티를 배열하는 방식으로 검출 및 교정한다.

주요포인트 꼭 알고 가야해!

데이터통신 합격은 당연! 고득점 합격이 목표!



[04강] 에러제어와 프로토콜

JODAEHO

❖ 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 보내올 때까지 기다리는 방식
- 연속적 ARQ : 한블록씩이 아니라 연속적으로 블록을 전송하는 방식
 - 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

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

❖ 후진 에러 수정(BEC)

- 송신측에서 전송한 프레임들 중 오류가 있는 프레임을 검출하여 오류가 발생되면 송신측에 오류가 있음을 통보하고 다시 재전송 받는 방식으로 역채널이 필요하다

주요포인트 꼭 알고 가야해!

[ACK]
긍정응답
[NAK]
부정응답



데이터통신 합격은 당연! 고득점 합격이 목표!



[04강] 에러제어와 프로토콜

JODAEHO

4. 통신 프로토콜의 개념

❖ 프로토콜(Protocol)의 정의

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

❖ 프로토콜의 기본적 구성 요소

- 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

❖ 프로토콜의 기능

- 단편화, 재합성, 캡슐화, 다중화, 동기화, 순서제어, 연결제어, 흐름제어, 오류제어, 주소결정, 전송서비스

❖ 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.

개체(서비스제공자)

SAP

개체(서비스사용자)

확인, 응답 →

← 요구, 지시

주요포인트 꼭 알고 가야해!

SNA[System Network
Architecture]
망구성제어기준

[SAP]
Service Access Point
서비스 접근 점
[서비스 프리미티브]
OSI 기본 참조 모델에서, 서비스 이용자와 제공자 간의 상호 동작 시 서비스를 제공하는 수단이 아니라 제공하는 서비스만을 주안점으로 하여 규정한 추상적 개념.
요구, 지시, 응답, 확인의 4 종류가 있다.



데이터통신 합격은 당연! 고득점 합격이 목표!



[04강] 에러제어와 프로토콜

JODAEHO

❖ 문자 위주 전송 프로토콜

- 반이중 통신 방식에 사용, P2P 또는 Multipoint에서 이용, 효율성 나쁨
- 데이터 길이 제한, Stop & Wait ARQ 방식 사용
- **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)
 - 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 위치서 유도
- ISO에서 현재 세계적으로 널리 사용되는 표준
- **HDLC(High-level Data Link Control) 프로토콜이 대표적**
- **SDLC, ADCCP**
- 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**

주요포인트 꼭 알고 가야해!

[프로토콜의 방식]

BSC : 문자(Character) 방식의 프로토콜(Loop 방식이 불가능함)

DDCMP : 바이트(Byte) 방식의 프로토콜

SDLC : 비트(Bit) 방식의 프로토콜

HDLC : 비트(Bit) 방식의 프로토콜



데이터통신 합격은 당연! 고득점 합격이 목표!



[04강] 에러제어와 프로토콜

JODAEHO

6. HDLC 프레임 구성

❖ HDLC 국의 종류

- 주국, 보조국, 복합국

❖ HDLC 프레임 구조

[플래그 - 주소필드 - 제어필드 - 정보필드 - 프레임 검사 순서 - 플래그]

8 8 8 가변 16 8

- 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 들어 있는 부분
- 프레임 검사 부분(FCS : Frame Check Sequence) : 주소부, 제어부, 정보부의 내용이 오류없이 상대방에 정확히 전송되었는가를 확인

❖ HDLC 프로토콜의 특징

- 비트지향형의 프로토콜, 제어부의 확장이 가능, 데이터링크 계층의 프로토콜
- 단방향, 반이중, 전이중 모두사용가능한 BIT방식 프로토콜

❖ HDLC 데이터 전달모드

- 정규 응답모드(Normal Response Mode)
- 비동기 균형모드(Asynchronous Response Mode)
- 비동기 응답모드(Asynchronous Balanced Mode)

주요포인트 꼭 알고 가야해!

SDLC(Synchronous Data Link Control)

HDLC의 변형된 형식 사용, 명령과 응답은 루프 토폴로지를 제공, 루프 또는 원형 폴링 동작 수행 제공



데이터통신 합격은 당연! 고득점 합격이 목표!



[04강] 에러제어와 프로토콜

JODAEHO

주요포인트 꼭 알고 가야해!



[05강] OSI / 인터넷

1. OSI 7계층 참조모델
2. OSI 7계층의 구조
3. 표준안 제정 기구
4. 인터넷과 프로토콜
5. 인터넷의 주소체계



대한민국 대표 IT강사
조 대 호

여러분의 인생드라마는
지금 잘 진행되시나요?
그 드라마의 주인공은
바로 당신입니다.
이 세상 모든사람들은
당신을 위한 조연이랍니다.
당신의 드라마가
성공한 그날!
당신의 감사메시지 속에서
저의 이름이 불려지기를
기원합니다.

데이터통신 합격은 당연! 고득점 합격이 목표!



1. OSI 7계층 참조 모델

JODAEHO

gisa plus

❖OSI(Open System Interconnection) 참조 모델의 정의

- 개방형 시스템의 상호 접속을 위한 참조 모델
- 제정 : ISO(국제 표준화 기구)에서 1977년 이기종 컴퓨터 통신을 위한 구조 개발에 의해 탄생된 규정
- 서로다른 기종의 컴퓨터 시스템 간에도 연결하여 통신을 할 수 있도록 한 표준화된 규격

❖OSI 참조 모델의 목적

- 시스템 간의 통신을 위한 표준 제공
- 시스템 간의 통신을 방해하는 기술적인 문제들을 제거
- 단일 시스템의 내부 동작을 기술해야 하는 노력을 제거
- 시스템 간의 정보 교환을 하기 위한 상호 접속점을 정의
- 관련 규격의 적합성을 조성하기 위한 공통적인 기반 구성

❖OSI 참조 모델의 기본 요소

- 개방형 시스템 : 응용 프로세스 간의 통신을 수행할 수 있도록 통신 기능을 제공
- 응용 실체 : 응용 프로세스를 개방형 시스템상의 요소로 모델화한 것
- 접속 : 응용 실체 간을 연결하는 논리적인 통신 회선
- 물리 매체 : 시스템 간에 정보를 교환할 수 있도록 해주는 전기적인 통신 매체

주요포인트 꼭 알고 가야해!

데이터통신 합격은 당연! 고득점 합격이 목표!



[05강] OSI와 인터넷

JODAEHO

2. OSI 7계층 구조

JODAEHO

gisa plus

❖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	게이트웨이
6	표현 계층	암호화, 내용 압축, 형식 변환 등의 기능을 제공	
5	세션 계층	사용자와 전송계층간의 연결 연결을 확립. 유지. 단절시키는 수단을 제공	
4	트랜스포트 계층	링크 종점 간에 정확한 데이터 전송을 제공하며 상,하위계층간의 인터페이스 역할담당	
3	네트워크 계층	데이터교환기능을 제공하며 패킷단위 전송, 정보교환, 경로선택,의 기능, IP프로토콜	라우터
2	데이터 링크 계층	데이터 전송을 행하고 전송오류를 제어하며 동기화, 에러제어, 흐름제어 기능을 담당	브릿지
1	물리 계층	실제 회선 연결을 확립.유지.단절하기 위한 기계적, 전기적, 기능적, 절차적 특성 정의	리피터

주요포인트 꼭 알고 가야해!

[흐름제어(flow control)]
패킷수를 적절히 조절하여
전체시스템의 안정성을 기
하는 기능,오버플로우 예방



데이터통신 합격은 당연! 고득점 합격이 목표!



[05강] OSI와 인터넷

JODAEHO

3. 표준안 제정기구

JODAEHO

gisa plus

❖ISO(International Organization for Standardization) : 국제 표준화 기구

❖ITU(International Telecommunication Union) : 국제전기통신연합

- 구 CCITT(Consultative Committee for Integration Telegraph and Telephone)가 개편된 국제전기통신연합
- V 시리즈 : 아날로그(전화망) 통신에서 사용되는 인터페이스를 위한 권고안
- X 시리즈 : 데이터 통신에서 사용되는 인터페이스를 위한 권고안

❖ANSI(American National Standard Institute) : 미국 표준 협회

- 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(International Electrotechnical Commission) : 국제전기표준회의

- 전기통신분야의 규격을 통일하기 위한 국제기구

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

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

❖IETF(Internet Engineering Task Force) : 인터넷국제표준화기구

- 인터넷의 운영, 관리, 기술개발 및 인터넷의 표준안을 제정

주요포인트 꼭 알고 가야해!

[DTE/DCE 접속규격]

DTE와 DCE의 기계적, 전기적인 접속규격의 표준안은 EIA, ITU 등에서 제정한다.

[표준화 접속규격]

RS-232C, RS-449, V.24, X.21 등

[X.21]

동기식 전송을 위한 DTE와 DCE접속규격

[X.25]

DTE와 패킷신호변환장치간의 데이터 교환절차를 정의하며 패킷교환망에서 사용되는 네트워크 프로토콜로 계층화 되었다.

[X.75]

패킷교환망과 패킷교환망의 연결을 위한 망간접속을 위한 프로토콜 권고안



데이터통신 합격은 당연! 고득점 합격이 목표!



[05강] OSI와 인터넷

JODAEHO

❖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 ARPANET는 인터넷의 모체가 되는 네트워크

❖TCP(Transmission Control Protocol) / IP(Internet Protocol)

- 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

❖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

❖인터넷의 주소

- IP 주소 : 인터넷에 연결된 컴퓨터가 갖는 고유한 주소
- 현재 사용하는 IP주소 체계는 IPv4로 32비트 체계
- 차세대 IP주소 체계는 IPv6, 32비트에서 128비트로 확장

❖도메인주소

- 문자,숫자로 구성된 인간이 이해하기 편한형태의 주소
- 호스트명.기관명.기관성격.국가명 으로 구성된다.

주요포인트 꼭 알고 가야해!

[TCP/IP관련 프로토콜]

UDP
ICMP(인터넷제어메세지프로토콜)
ARP(주소결정프로토콜)

발행	도메인 및 의미
NIC (미국)	com(영리단체) org(비영리단체) net(네트워크)
KRNIC (한국)	co.kr(영리단체) or.kr(비영리단체) ne.kr(네트워크) pe.kr(개인) ac.kr(전문대이상) hs.kr(고등학교) ms.kr(중학교) es.kr(초등학교) sc.kr(특수학교)



데이터통신 합격은 당연! 고득점 합격이 목표!



[05강] OSI와 인터넷

JODAEHO

5. 인터넷의 주소체계(IPv4)

❖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

❖DNS(Domain Name System)

- 도메인 이름을 IP주소로 상호 변환하는 기능

❖TCP/IP에 운용되는 인터넷 서비스

- 핑(Ping)
 - 사용자가 접속하려는 호스트가 운영되고 있는지를 확인할 수 있는 서비스
- 핑거(Finger)
 - 특정 인터넷 사용자의 사용한 시간, 접속 여부 등에 관한 정보를 파악할 수 있는 서비스
- 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) 프로토콜
 - E-mail을 수신하기 위한 POP3(Post Office Protocol3) 프로토콜

주요포인트 꼭 알고 가야해!

[D 클래스]
멀티캐스트용



[웹 2.0]

데이터의 소유자나 독점자 없이 누구나 손쉽게 데이터를 생산하고 인터넷에서 공유할 수 있도록 한 사용자 참여 중심의 인터넷 환경

데이터통신 합격은 당연! 고득점 합격이 목표!



[05강] OSI와 인터넷

JODAEHO

주요포인트 꼭 알고 가야해!



[06강] 정보통신망

1. 데이터교환방식
2. LAN / MAN / WAN / VAN
3. LAN의 토폴로지
4. ISDN
5. 이동통신망



대한민국 대표 IT강사
조 대 호

여러분의 인생드라마는
지금 잘 진행되시나요?
그 드라마의 주인공은
바로 당신입니다.
이 세상 모든사람들은
당신을 위한 조연입니다.
당신의 드라마가
성공한 그날!
당신의 감사메시지 속에서
저의 이름이 불려지기를
기원합니다.

데이터통신 합격은 당연! 고득점 합격이 목표!



1. 데이터 교환 방식

JODAEHO

gisa plus

❖회선 교환 방식

- 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

❖패킷 교환 방식

- 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
- 장애발생시 대체경로 선택이 가능하다.
- 가상회선방식(연결지향형)과 데이터그램방식으로 세분화 됨
- 플러딩(flooding) : 각 노드에서 들어온 패킷을 다른 모든 링크로 복사하여 전송하는 형태
- PAD : 패킷을 분해 조립하는 장치

❖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

주요포인트 꼭 알고 가야해!

[데이터그램]

정해진 경로가 없고 융통성 및 신뢰성이 좋다.
송신수 순서와 수신시 순서가 달라질수 있다.

[가상회선방식]

논리적경로를 확보하여 전송하며 많은양의 데이터전송시 좋으며 노드 지연이 발생될수 있다.
순서가 일치하므로 순서제어가 필요없다.



데이터통신 합격은 당연! 고득점 합격이 목표!



[06강] 정보통신망

JODAEHO

❖LAN의 개요

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

❖LAN의 특징

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

❖MAN

- 도시간 통신망

❖WAN(Wide Area Network)의 정의

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

❖VAN(Value Added Network)의 개념 및 정의

- 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

주요포인트 꼭 알고 가야해!

[이더넷(Ethernet)]

대표적인 버스구조방식의 LAN이며 데이터전송을 위해 CSMA/CD방식을 사용한다.

10 BASE 5
(10Mbps, 500m)

[LAN에서 사용되는 매체 액세스 제어(Access Control)기법]

TOKEN-BUS
TOKEN-RING
CSMA/CD(충돌방지법)



데이터통신 합격은 당연! 고득점 합격이 목표!



[06강] 정보통신망

JODAEHO

3. LAN의 토폴로지

❖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

❖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

주요포인트 꼭 알고 가야해!

데이터통신 합격은 당연! 고득점 합격이 목표!



[06강] 정보통신망

JODAEHO

❖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
- B-ISDN(Broadband-ISDN, 광대역 ISDN) : 영상회의, 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

❖ISDN의 채널

- A채널 : 아날로그 가입자선의 정보채널
- B채널 : 64Kbps이하의 PCM화된 디지털 음성이나 정보용 채널
- C채널 : A채널과 관련된 혼합 액세스 채널(8~16Kbps)
- D채널 : 16 또는 64Kbps의 속도로 회선교환방식을 위한 신호용 채널
- H채널 : B채널을 통해 제공하는 모든 방식의 정보 전송을 보다 고속으로 제공

❖ISDN 서비스

- 베어러 서비스(Bearer Service) : 기본전송위주의 통신서비스이며 OSI의 하위 계층에 해당되는 정보 전달 서비스로 음성, 음향, 영상등을 실시간으로 전달
- 텔레 서비스(Tele-Service) : 베어러 서비스를 기본으로 한 응용 위주의 서비스로 OSI참조 모델의 상위계층에 해당하며 정보전달기능과 정보처리기능이 결합된 서비스
- 부가 서비스(Supplementary Service) : 베어러, 텔레서비스 이외의 사용자 편의를 위한 서비스

❖ATM(비동기 전송모드)

- 각채널을 53byte로 분할 후 48byte마다 분리하여 5byte의 헤더를 사용
- 연결형 비동기전송모드로 B-ISDN의 핵심기술이다.

주요포인트 꼭 알고 가야해!

[1차군(BRI)접속]

2B+D 채널로 주로 소형점포나 대리점에서 사용

[2차군(PRI)접속]

보다 큰 대역폭으로 제공하는 회선으로 복미방식의 23B+D 채널과 유럽방식의 30B+D 채널로 구분된다.



[ADSL(Asymmetric Digital Subscriber Line)]

비대칭 디지털 가입자 회선

데이터통신 합격은 당연! 고득점 합격이 목표!



[06강] 정보통신망

JODAEHO

5. 이동 통신망

❖셀룰러(Cellular) 시스템

- 서비스 지역의 제한과 가입자 수용 용량의 한계를 극복하기 위하여 제안된 개념
- 주파수 재사용 : 주파수를 재활용하는 방법으로 같은 주파수를 다른셀에서 사용
- 셀 분할 기법 : 여러 개의 작은 셀로 나누어 동일주파수를 이용하는 방식
- 핸드오프(오버) : 통화 중인 가입자가 새로운 기지국 서비스 지역으로 진입할 때도 통화의 단절없이 계속 통화가 가능(Hard핸드오버, Soft핸드오버, Softer핸드오버)
- 로밍 : 이동전화 가입자가 타 교환국에 있어도 이동전화 서비스를 받을 수 있음)의 특징이 있음

❖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 누구하고나 어떠한 형태의 통신도 가능함

❖IMS(IP Multimedia Subsystem)

- 고정전화망이나 이동통신망등 인터넷전화에 쓰이는 프로토콜인 SIP로 통합하여 멀티미디어 서비스를 실현시키는 방식

❖위성통신

- 적도상공 약 36,000Km상공의 정지위성을 이용하여 고주파수대의 전파를 이용, 초고속 전송이 가능하며 지형에 관계없이 고른 통신이 가능하다.

주요포인트 꼭 알고 가야해!

[WAP]

[Wireless Application Protocol]

셀룰러 폰이나 무선 호출기 등과 같은 무선 장치들이 전자 우편, 웹 뉴스 그룹 및 IRC 등의 인터넷 액세스에 사용될 수 있는 방법을 표준화하기 위한 통신 프로토콜

[3개의 위성]

전세계 모든지역이 위성으로 통신이 가능하려면 지구상공에 3개이상의 정지위성을 배치하면 된다.

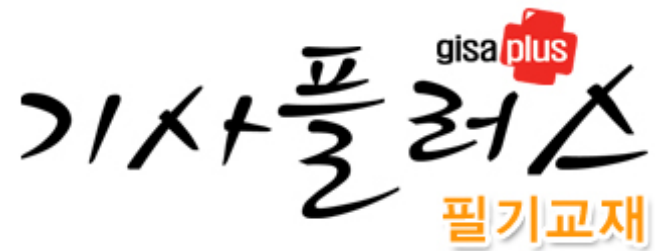


데이터통신 합격은 당연! 고득점 합격이 목표!



[06강] 정보통신망

JODAEHO



정보처리기사/산업기사 **필기**

소 프 트 웨 어 공 학

시스템분석및설계



주요포인트 꼭 알고 가야해!

(GO) [목차] 소프트웨어공학 & 시스템분석, 설계

- ❖ 시스템과 코드 설계
- ❖ 입력과 출력 설계
- ❖ 프로세서와 프로그래밍 설계
- ❖ 시스템의 평가와 문서화
- ❖ 소프트웨어 공학 및 IPT 기법
- ❖ 구조적 분석과 객체 지향 분석
- ❖ **소프트웨어 프로젝트 관리(기사)**
- ❖ **소프트웨어 공학의 발전적 주제(기사)**

정보처리기사	문항	정보처리산업기사
데이터베이스	20	데이터베이스
전자계산기구조	20	전자계산기구조
운영체제	20	운영체제
데이터통신	20	정보통신개론
소프트웨어공학	20	시스템분석, 설계
기사합계	100	산업기사합계

합격
비법대한민국 대표 IT강사
조 대 호

여러분의 인생드라마는
지금 잘 진행되시나요?
그 드라마의 주인공은
바로 당신입니다.
이 세상 모든사람들은
당신을 위한 조연입니다.
당신의 드라마가
성공한 그날!
당신의 감사메시지 속에서
저의 이름이 불려지기를
기원합니다.

정보처리기사/산업기사 합격은 당연! 고득점 합격이 목표! (GO)

주요포인트 꼭 알고 가야해!

(GO) [01강] 시스템과 코드설계

1. 시스템의 정의
2. 시스템의 기본요소
3. 코드설계
4. 코드의 종류1
5. 코드의 종류2
6. 오류검사방법

합격
비법대한민국 대표 IT강사
조 대 호

여러분의 인생드라마는
지금 잘 진행되시나요?
그 드라마의 주인공은
바로 당신입니다.
이 세상 모든사람들은
당신을 위한 조연입니다.
당신의 드라마가
성공한 그날!
당신의 감사메시지 속에서
저의 이름이 불려지기를
기원합니다.

정보처리기사/산업기사 합격은 당연! 고득점 합격이 목표! (GO)

❖시스템의 정의

- 어떤 목적을 위하여 하나 이상의 기능 요소가 상호 관련하여 유기적으로 결합된 것으로 체계, 계통, 조직, 제도, 방식 등을 총칭하는 말이다.
- 예정된 기능을 수행하기 위하여 설계된 상호작용을 갖는 요소의 유기적 집합체이다.
- 어떤 목적을 위하여 하나이상의 기능요소가 상호 관련하여 유기적으로 결합된 것이다.
- 공통의 목적에 의하여 공통의 목적에 기여할 수 있는 많은 이질부분으로 구성되는 복잡한 단일체이다.

❖시스템의 특성

- 목적성
- 자동성
- 제어성
- 종합성

❖시스템 분석가

- 사용자와 개발자간의 중재자 입장, 개발에 관한 총책임자.
- 기술적인 측면보다는 관리적인 측면이 더 중시됨

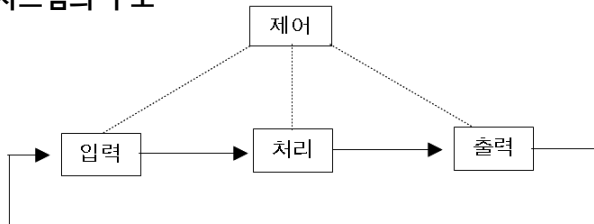
❖시스템 분석가의 자질

- 기업의 목적을 정확히 이해하고 문제의 타당성 여부를 검사하는 능력
- 기업의 환경조사, 현상 분석의 능력
- 시스템의 기능 분석 및 기본 설계 능력
- 컴퓨터 기술과 시스템의 운영, 관리 기법의 이해 능력

주요포인트 꼭 알고 가야해!

2. 시스템의 기본 요소

❖시스템의 구조



피드백

- 입력(Input) : 처리를 위해 주어지는 조건 및 자료
- 출력(Output) : 처리된 결과
- 처리(Process) : 입력된 자료를 조건에 따라 처리, 결과 출력을 위한 변환 행위
- 피드백(Feedback) : 처리된 결과의 측정, 분석 후 목표치 도달 여부 확인과 만족스럽지 못한 결과는 다시 조정하는 행위
- 제어(Control) : 시스템의 기본요소들이 각 과정을 올바르게 행해진지 감독하는 요소

❖시스템 설계의 순서



- 개요 설계 : 분석 결과에 따라 사용자 입장에서 시스템 전체를 개괄적으로 설계하는 것
- 상세 설계 : 전산화하기 위한 구체적인 작업 순서 결정, 컴퓨터 및 입출력 기기를 사용하여 이미 결정된 단계에 맞게 시스템 제작 방법을 설계

주요포인트 꼭 알고 가야해!

❖코드의 개념

- 데이터의 수집이나 분류, 집계 등을 용이하기 위해 사용되는 기호나 부호를 코드라 한다.

❖코드의 기능

- 식별 기능
- 분류 기능
- 배열 기능

❖코드의 특징

- 자료의 분류집계를 쉽게 한다.
- 자료의 구별과 특정자료의 추출이 용이하다.
- 취급하기 쉽고 체계가 있게 설계하여야 한다.
- 확장성이 있게 설계하여야 한다.
- 컴퓨터 처리에 적합하도록 1:1로 대응시킨다.
- 자료의 표현은 단순하고 짧게 한다.

❖코드 설계 순서

- ① 코드 대상 선정
- ② 사용범위와 사용기간의 결정
- ③ 코드설계와 체크
- ④ 코드의 번역, 코드표 작성
- ⑤ 코드 파일 작성
- ⑥ 코드 파일과 코드표 관리

주요포인트 꼭 알고 가야해!



4. 코드의 종류1

❖Group Classification code(그룹 분류식 코드)

- 코드화 대상 항목을 소정의 기준에 따라 대분류, 중분류, 소분류로 구분하고, 각 그룹 안에서 순차 번호를 배정하여 코드화
- 장점 : 각 자리수마다 특정의 의미를 나타낼 수 있어 구분별 집계 용이하며 추가가 자유스럽고 전산 처리에 가장 적당하다.
- 단점 - 자리수가 많이 소요된다.

❖Decimal code(10진 코드)

- 도서관에서 도서 정리를 목적으로 만든 것으로 좌측부는 그룹분류에 따르고 우측은 10진수의 원칙에 따라 세분화하는 코드
- 10진 분류해서 대, 중, 소로 구분
- 주로 도서관에서 사용
- 자료의 추가가 용이, 무한하게 확대가 가능
- 자리수가 많아지고 기계 처리에 불편한 코드

❖Mnemonic code(연상식 코드, 연상 기호 코드)

- 실제 자료를 연상하기 쉬운 약어나 기호로써 코드화
- 장점 - 해독이 용이하고 코드의 변경이 간단하다.
- 단점 - 칼럼 수가 많고 분류가 어렵다.

제품	코드
냉장고(235ℓ)	RF-235
형광등(30W 220V 흰색)	K-30-220-W
텔레비전(17인치 컬러)	T-17
텔레비전(25인치 컬러)	T-25-C

주요포인트 꼭 알고 가야해!

❖ Significant digit code(표의 숫자식 코드)

- 코드화 대상 항목의 길이, 넓이, 부피, 무게 등을 나타내는 문자, 숫자 혹은 기호를 그대로 코드로 사용하는 코드
- 중량, 용량, 면적, 거리, 광도 등 물리적 수치를 직접 code에 적용

❖ Sequence code(일련 번호식 코드)

- 자료의 발생순서나 일정 기준에 따라 일련번호를 부여하는 방법으로 항목수가 적고 변경이 없는 경우에 사용
- 장점 - 간편하고 마지막 추가가 용이하다.
- 단점 - 중간 추가가 불가능하며 분류 체계가 없고 항목수가 많으면 코딩이 어렵다.

❖ Block code(구분식 코드)

- 공통성이 있는 것끼리 묶어 블록을 형성하고 블록 내에서 일련번호를 부여하는 코드
- 장점 - 짧은 행수로 구분을 표시할 수 있고 예비코드를 사용할 수 있어서 추가가 매우 용이하다.
- 단점 - 구분마다 예비코드를 가지고 있어서 기계처리가 복잡하다.

❖ Letter type code(약자식 코드)

- 관습상 제도상 널리 사용하는 약자를 코드로 사용하는 것으로 세계 공통으로 사용
- ex) kg, km

❖ Numerical Alphabetical code

- 코드화 대상 항목을 예정된 방식에 따라 문자를 숫자로 치환하는 코드
- ex) A 01, B 02 ~ Z 26

주요포인트 꼭 알고 가야해!

Cryptic Code(암호화 코드)

외부나 관계자 외에 알리면 난처하게 될 경우

Final digit code(말미식 코드)

마지막 자리에 의미를 부여하여 사용



❖ 프로그램에 의한 검사

- Limit check : 입력 자료의 어떤 항목 내용이 논리적으로 정해진 범위내에 있는가 체크
- Data Count check : 컴퓨터로 처리할 데이터의 개수와 컴퓨터로 처리한 데이터의 개수가 같은지의 여부를 검사하는 체크 방법
- Batch total check (=sum check) : 컴퓨터 입력 단계의 체크에서 입력 정보의 두 가지 이상이 특정 항목의 합계 값과 같다는 것을 알고 있을 때, 계산 결과가 같은지를 체크
- Balanced check : 컴퓨터 입력 단계의 체크(check) 중 입력 정보의 두 가지 이상이 특정 항목의 합과 같다는 것을 알고 있을 때, 컴퓨터를 이용해서 계산한 결과와 분명히 같은지를 체크(경리 장부 처리 시 차변, 대변의 한계 값을 체크하는데 사용하는 방법)
- Logical check(Validity check, 논리 검사) : 주어진 논리조건에 맞는 자료인지 검사
- 대조 check : 입력 data를 data code 표와 대조하여 그 안에 있는 것과 같은지를 확인
- Sequence check : 입력되는 Data의 배열이 정해진 순서대로 정돈되어 있는가를 Check
- Numeric check : 숫자 항목으로 정해진 곳에 숫자 아닌 문자가 입력되어 있는지 Check
- Blank check : blank로 있어야 할 Field가 확실히 blank로 있는지 check
- Label check : 파일들을 나타낼 때 식별하기 위한 검사
- Echo check(반향 검사) : 수신한 데이터를 송신 단말기에 되돌려 보내 원래와 비교검사
- Mode check(모드 검사) : 입력되는 자료가 숫자, 문자, 영숫자 등 주어진 타입인지 검사

❖ Check digit check

- check digit가 붙어 있는 code가 입력될 때 computer에서 규정에 따라 계산하여 error를 발견하는 검사법으로 전산처리에 가장 적합하다. 은행의 계좌번호나 주민번호 등에 사용되고 있다.
- 주민번호끝자리판별공식 : 11-(가중치(234567892345)의 곱의 총합을 11로 나눈 나머지)

주요포인트 꼭 알고 가야해!

[오류의 종류]

•오자 에러(Transcription error) : 하나의 숫자나 문자의 오기로 발생한 에러
ex) 1973 => 1943

•전위 에러(Transposition error) : 입력 시 좌우 자리를 바꾸어 발생한 에러.
ex) 1973 => 1937

•이중 에러(Double error) : 전위 에러가 2개 이상 발생한 에러

•생략 에러(Omission error) : 입력 시 한자리를 빠뜨리고 기록한 에러
ex) 34278 => 3428

•랜덤 에러(Random error) : 위의 3개중 2개 이상 겹친 error



주요포인트 꼭 알고 가야해!

[GO] [02강] 파일, 프로세스설계

1. 입력설계
2. 출력설계
3. 파일매체설계
4. 색인순편성방법
5. 랜덤, 리스트편성방법
6. 프로세스설계

합격
비법대한민국 대표 IT강사
조 대 호

여러분의 인생드라마는
지금 잘 진행되시나요?
그 드라마의 주인공은
바로 당신입니다.
이 세상 모든사람들은
당신을 위한 조연입니다.
당신의 드라마가
성공한 그날!
당신의 감사메시지 속에서
저의 이름이 불려지기를
기원합니다.

정보처리기사/산업기사

합격은 당연! 고득점 합격이 목표!

[GO]**1. 입력설계**

JODAEHO

gisa plus

❖입력 매체와 형식

- 입력 형식에서 사용되는 입력 매체가 무엇이나(일괄 처리 방법과 온라인 처리 방법)에 따라 크게 좌우 된다.
- 마크 카드(Mark Card) 이용
 - OMR : 마크에 의한 방식
 - OCR : 문자에 의한 방식
 - MICR : 특수 잉크로 기록된 문자를 입력

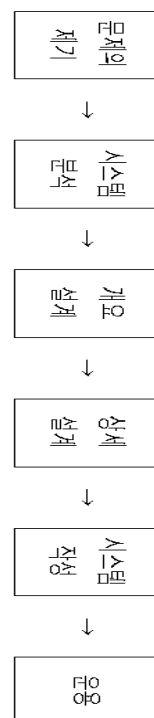
❖입력 정보의 설계 순서

- 입력 정보의 발생에 관한 설계 : 입력정보의 명칭, 작성목적, 발생자와 발생장소, 발생 방법 및 형태, 발생 주기 및 시기, 발생건수
- 입력 정보의 수집에 관한 설계 : 수집가 및 수집 장소, 수집방법 및 형태, 수집경로, 수집주기 및 시기
- 입력 정보의 매체화에 관한 설계 : 입력 형식의 선택, 입력 매체 및 장치의 선택, 매체화 기기, 매체화 담당자, 장소, 건당 문자 수, 매체화 주기 및 시기
- 입력 정보의 투입에 관한 설계 : 입력 장치, 투입 주기 및 시기
- 입력 정보의 내용에 관한 설계 : 항목명, 항목의 배열, 항목의 자리 수, 항목의 문자 구분

❖입력 방식

- 집중 매체화 입력 방식 : 발생한 데이터를 전표상에 기록하고, 일정한 시간 단위로 일괄 수집하여 입력매체에 수록하는 입력 형식
- 분산 매체화 입력 방식 : 입력 정보의 매체화를 정보가 발생한 장소에서 입력하고, 일정 주기로 수집하여 투입하는 입력 방식
- 회귀 자료 시스템(Turn Around System) : 입력된 자료가 처리되어 일단 출력된 후 이 용자를 거쳐 다시 재입력되는 방식으로, 공과금, 보험료 징수 등의 지로용지를 처리하는데 사용되는 입력방식

주요포인트 꼭 알고 가야해!



정보처리기사/산업기사

합격은 당연! 고득점 합격이 목표!

[GO]**[02강] 파일, 프로세스설계**

Slides 10

❖출력 매체와 형식

- 인쇄 출력 시스템
 - 라인 프린터(line printer)
 - 도트프린터(dot printer)
 - X-Y 플로터(plotter)
 - 레이저 프린터(laser printer)
 - 잉크젯 프린터
- COM(computer output microfilm) 시스템
 - 종이에 출력하는 대신 출력 정보의 형태나 글자를 자기 테이프로부터 마이크로필름에 수록하는 방식이다.

❖출력 정보의 설계 순서

- 출력 정보의 내용에 관한 설계 : 항목의 배열, 자리 수, 문자 구분, 집계 구분
- 출력 정보의 매체화에 관한 설계 : 출력 형식, 매체 및 장치의 선택, 출력 보조 장치, 출력 장소, 정보 기능, 문자 수, 배열 순서
- 출력 정보의 분배에 관한 설계 : 분배 책임자, 분배 방법 및 형태, 분배경로, 분배 주기 및 시기
- 출력 정보 이용에 관한 설계 : 출력 정보명, 출력 목적, 사용자와 이용 경로, 이용 주기 및 시기, 기밀성의 유무와 보존

❖입력정보설계순서와 출력정보설계순서 외우기

- 입력설계 : 발 → 수 → 매 → 투 → 내
- 출력설계 : 내 → 매 → 분 → 이

주요포인트 꼭 알고 가야해!

3. 파일매체설계

❖파일 매체의 설계

- 기능 검토 : 액세스 형태, 처리 방식, 데이터의 양, 작동의 용이성
- 종합 검토 : 매체의 선정, 매체수의 산출, 장치 대수의 산출
- 파일 설계의 순서
 - ① 파일작성의 목적 확인
 - ② 파일 항목의 검토
 - ③ 파일의 특성조사
 - ④ 파일매체의 검토
 - ⑤ 편성법 검토

❖파일 편성의 방법

- 순편성, 색인순편성, 랜덤편성, 리스트편성

❖순편성 방법(sequential organization)

- 장점
 - 속도가 빠르며, 처리비용이 절감된다.
 - 어떤 매체라도 순편성 파일의 기록 매체가 될 수 있다.
 - 기억 공간의 활용률이 높다.
- 단점
 - 파일 내에 새로운 레코드를 삽입, 삭제하는 경우 파일 전체를 새로운 매체에 복사해야 한다.
 - 복사에 따른 작업 시간과 비용의 낭비가 크다.
 - 특정 레코드 검색 시 효율이 떨어진다.
- 대표적인 기록매체
 - 자기 테이프 : 순차적 기억 매체,
 - 자기 디스크, 자기 드럼 : 직접 접근 기억 매체

주요포인트 꼭 알고 가야해!

❖ 색인 순편성 방법(indexed sequential organization)

- 순편성 파일의 장점을 유지하면서 순차적인 접근으로 문제를 해결하고자 하는 방법으로 순차적 처리와 임의 처리가 모두 가능하다.
- 색인 순편성 파일의 구성
 - 기본 데이터 구역(prime data area) : 실제 데이터가 기록된 부분
 - 인덱스 구역(index area) : 기본 데이터 구역에 관한 색인이 기록된 부분
 - 트랙 색인 구역(track index area)
 - 실린더 색인 구역(cylinder index area)
 - 마스터 색인 구역(master index area)
 - 오버플로우 구역(overflow area) : 새로 추가되는 레코드가 많아서 데이터 구역 내에 다 기록할 수 없을 때 사용하는 부분
 - 독립 오버플로우 구역(independent overflow area)
 - 실린더 오버플로우 구역(cylinder overflow area)
- 장점
 - 순차적 처리와 임의 처리가 가능하므로 파일을 융통성 있게 처리할 수 있다.
 - 파일 내에 레코드를 추가하거나 삭제할 경우 파일 전체를 복사할 필요가 없다
- 단점
 - 기억 공간의 낭비가 크다.
 - 접근 시간이 느리다.
 - 오버플로우 되는 레코드가 많아지면 사용 중에 파일을 재편성해야 하는 문제점이 발생한다.
- 사용되는 기록 매체 : 임의 처리가 가능한 매체 즉, 자기 디스크나 자기 드럼 등
- 응용 파일 : 부품 원가 파일, 코드 마스터 파일

주요포인트 꼭 알고 가야해!

❖ 랜덤 편성 방법(random organization)

- 파일 내의 레코드들이 어떤 일정한 순서를 유지할 필요가 없는 임의 처리에 사용된다.
- 특정 항목을 키로 하여 주소를 계산해 내고, 이 주소에 레코드를 기억시키는 방법이다.
- 키 주소 변환 방법
 - 숫자 분석법(digit analysis), 접지 변환법(folding transformation)
 - 지수 변환법(radix transformation), 나눗셈 변환법(division transformation)
 - 순서 유지 변환법(sequence-maintaining transformation)
 - 중간수 제곱 변환법(mid-square transformation)
- 장점 : 레코드를 주소 계산에 의하여 직접 처리할 수 있다.
 - 어떤 레코드라도 평균 접근 시간 내에 검색이 가능하다.
 - 운영 체제에서 키 주소 변환을 자동적으로 해준다.
- 단점 : 키 변환을 하므로 시간이 지연된다.
 - 키 변환 방법에 따라 공백(space)이 많으므로 효율이 떨어진다.
- 대표적인 매체 : 임의 접근이 가능한 자기 디스크나 자기 드럼
- 응용 파일 : 거래 작업이 빈번한 재고 관리 파일, 계약 마스터 파일

❖ 리스트 편성 방법(list organization)

- 데이터 레코드들이 물리적으로 떨어져 있으나 데이터 레코드에 포함되어 있는 포인터가 다음 자료의 주소를 지시함으로써 관계를 유지하는 파일이다.
- 장점 : 레코드의 추가, 삭제의 경우 포인터 값만 바꾸어 주면 되므로 파일 전체를 복사할 필요가 없다.
 - 포인터 값의 변경으로 레코드의 삽입, 삭제가 용이하다.
- 단점 : 파일의 구조가 복잡하며 처리 효율이 떨어지고 기억 장소의 낭비가 크다.
- 대표적인 매체 : 임의 접근이 가능한 매체, 자기 디스크나 자기드럼
- 응용 파일 : 부품 전개 파일, 계약, 출고, 입금 명세 파일 등

주요포인트 꼭 알고 가야해!

[주소를 결정하는 방법]

- ① 직접 매핑(direct mapping) : "키값 = 주소"의 관계를 유지
- ② 디렉토리 조사(directory lookup) : "키값, 주소" 쌍으로 된 디렉토리를 유지
- ③ 해싱(hashing) : hashing function 이용



❖ 프로세스 설계의 기본 원칙

- 입력 정보로 올바른 출력 정보를 얻을 때까지의 기본 처리 과정을 명확하게 표현
- 작업 절차에 맞게 프로세스 흐름도를 먼저 작성하고, 각각의 프로세스 논리를 설계
- 표준 패턴을 적절히 조합하여 처리 절차에 따라 순서를 나열하는 방식으로 도표 작성

❖ 흐름도의 종류

- Block Chart : 컴퓨터나 사용 장치의 기능적 관계를 표현한 흐름도
- System Flowchart : 자료 발생부터 결과를 얻기까지 시스템의 전 과정을 나타낸 흐름도
- Process Flowchart : 컴퓨터로 문제를 처리하기 위한 프로그램의 시스템적 흐름도
- Program Flowchart : 문제의 논리적인 흐름에 따라 프로그래밍하기 위해 작성것
- Progress Flowchart : 작업 진행 내용을 각 공정별로 따로 작성한 흐름도

❖ 데이터 프로세스의 기본 형식

- 정렬(sort) : 비순서적인 자료 파일을 분류키에 따라 순서적(오름차순, 내림차순)으로 나열하는 것
- 병합(Merge) : 같은 파일 형식을 갖는 2개 이상의 파일을 하나의 파일로 통합하는 처리
- 대조(Match) : 파일내의 자료와 대조 파일에 있는 자료와 같은 것들만 고르는 것을 말하며,
- 갱신(Update) : 마스터 파일의 데이터를 트랜잭션 파일에 의해 추가, 삭제, 교환하여 새로운 마스터 파일을 작성하는 처리 패턴
- 추출(Extract) 또는 정보 검색(information retrieval) : 자료 파일 내에 있는 자료를 조건에 맞는 것만 추출해 내는 것
- 분배(Distribution) : 어떤 특정한 조건을 부여하여 조건을 만족시키는 정보와 만족시키지 못하는 정보로 분리하여 처리하는 패턴
- 매체변환(conversion) : 입력 매체인 종이테이프 또는 펀치 카드상의 데이터를 자기 디스크에 수록하는 처리는 프로세스의 표준 패턴

주요포인트 꼭 알고 가야해!

[프로그래밍 절차]

타당성 조사

문제 분석

입출력 설계

Memory Map 작성

Flow Chart 작성

Program Coding

Program 입력

Compile 및 Error 체크

실제 Data로 결과 출력

주요포인트 꼭 알고 가야해!

GO

[03강] 시스템과 소프트웨어

1. 시스템평가
2. 문서화
3. 소프트웨어의 중요성
4. 소프트웨어의 특징
5. SDLC(소프트웨어 생명주기)

합격
비법대한민국 대표 IT강사
조 대 호

여러분의 인생드라마는
지금 잘 진행되시나요?
그 드라마의 주인공은
바로 당신입니다.
이 세상 모든사람들은
당신을 위한 조연이랍니다.
당신의 드라마가
성공한 그날!
당신의 감사메시지 속에서
저의 이름이 불려지기를
기원합니다.

❖기능, 성능, 신뢰성의 평가

- 기능의 평가 : 시스템이 요구되는 기능을 정확하게 수행하는지를 평가한다.
- 성능의 평가
 - 시스템이 운영 계획에서 마련한 운영 스케줄(시간)대로 수행할 수 있는지를 평가
 - CPU의 속도 및 기억 용량, 파일의 편성과 액세스 방식, 입출력 장치의 처리 속도
- 신뢰성 평가
 - 시스템이 얼마나 고장 없이 많은 작업을 정확하게 처리하여 주는가를 평가한다.
 - 신뢰성의 평가는 평균 고장시간(MTBF)과 평균 수리 시간(MTTR)으로 산출한다.
 - 예를 들어 어떤 시스템의 운영 기간이 다음과 같다고 하자.

가동중	고장중	가동중	고장중	가동중	고장중
a1	b1	a2	b2	a3	b3

- MTBF(Mean Time Between Failure) : 평균 고장 시간
고장 완료 시점으로부터 다음 고장 발생 시점까지 작업이 가동 중인 시간의 평균치

$$MTBF = \frac{(a_1 + a_2 + a_3)}{3}$$

- MTTR(Mean Time To Repair) : 평균 수리 시간
시스템의 고장으로 인해 작업하지 못한 시간의 평균치

$$MTTR = \frac{(b_1 + b_2 + b_3)}{3}$$

- 신뢰도 : 전체 시간 중에서 가동 중인 시간의 비율(가용도라고도 함)

$$\text{신뢰도} = \frac{MTBF}{MTBF + MTTR} = \frac{a_1 + a_2 + a_3}{a_1 + a_2 + a_3 + b_1 + b_2 + b_3} = \frac{\text{총가동시간}}{\text{총운영시간}}$$

주요포인트 꼭 알고 가야해!

2. 문서화

❖문서화의 필요성

- 문서화는 소프트웨어 개발에 관련된 모든 행위를 문서로 만들어 두는 것을 말하며, 다음과 같은 목적으로 한다.
- 업무 담당자가 교체되는 경우에 새로운 담당자가 기존 작업에 대한 내용을 쉽게 파악할 수 있도록 하기 위해 반드시 필요하다.
- 시스템 사용 도중에 변경이나 유지 보수가 용이하도록 사전에 준비해 두어야 한다.

❖문서화 규칙

- 프로그램 자체로서 보고서 기능을 할 수 있도록 한다.
- 다른 프로그램을 작성하기 위한 지침서로도 사용할 수 있도록 해야 한다.
- 형식을 표준화하여 새로운 프로그래머가 개발 작업을 쉽게 할 수 있도록 한다.
- 전체 업무에 대한 개략, 상세 설계와 함께 모든 내용을 문서화의 표준 규칙에 따라 누구나 알아보기 쉽게 작성해야 한다.
- 표준화를 통해 수정 및 관리에 따른 유지 보수가 쉬워지도록 한다.
- 지침서 및 흐름도를 규정하여 통일성 있게 프로그래밍할 수 있도록 한다.

❖문서화의 이점

구 분	효 과
관 리 자	작업 계획 수립이 쉬워지고 작업 관리가 쉬워진다 인원 투입 계획의 수립이 쉬워진다 작업의 분업 및 외주 관리가 쉬워진다
개 발 자	소프트웨어 시스템에 대한 이해가 쉬워진다 시스템 분석, 설계를 효율적이고 쉽게 할 수 있다 프로그램의 작성이 쉬워지고 에러를 방지할 수 있다 에러의 수정이 쉬워져 프로그램의 개선 및 보수가 쉬워진다

주요포인트 꼭 알고 가야해!

[문서화의 표준화]

- 분석, 설계, 구현, 테스트, 유지보수, 활동을 하면서 나타나는 산출물을 문서화
- 문서화 표기의 표준화는 널리 알려진 도표나 표기법을 사용하여 문서화

❖ 소프트웨어 위기(Software Crisis)

- 하드웨어의 기술의 급속한 발전과 범용 컴퓨터의 광범위한 보급은 다음과 같은 문제를 야기하였으며 이는 소프트웨어 엔지니어들에게 위기의식을 불러 일으켰다.
- 하드웨어의 기술은 그 기능을 이용하는 소프트웨어의 개발 능력의 발전 속도보다 훨씬 빠르게 발전한다.
- 새로운 소프트웨어를 요구하는 시장의 수요를 감당할 수 없다.
- 기존 소프트웨어의 유지보수가 힘들어진다.

❖ 소프트웨어 공학

- 하드웨어의 급속적인 발전은 소프트웨어에 대한 사용자의 요구가 증가되고 사용자의 불만족 등 여러 가지 문제점들이 증가하게 되었다.
- 소프트웨어공학은 이러한 소프트웨어의 문제점 혹은 위기를 극복하기 위해 마련된 학문이며 소프트웨어의 품질과 생산성을 향상시키고 작업 만족도를 증대시키는데 그 목적을 두고 있다.

❖ 소프트웨어 공학의 요소

- 프로세스(Process)
 - 프로그램 개발 시 사용되는 방법과 도구가 사용되는 순서를 규정한 것
 - 프로그램에 투여되는 인원 및 기타 환경에 대해서 정의한 것
- 방법(Methodology)
 - 소프트웨어 개발에 사용되는 기술적인 방법을 제공
 - 예) 프로세스 중심 방법론, 자료 구조 중심 방법론, 객체 지향 방법론
- 도구(Tool)
 - 프로그램 개발 시 사용되는 여러 가지 방법을 일정한 순서에 의해 자동으로 처리해 주는 기능
 - 예) CASE(Computer Aided Software Engineering)

주요포인트 꼭 알고 가야해!

❖ 소프트웨어란

- 실행 시 요구되는 기능과 성능을 제공해 주는 컴퓨터 명령어들의 모임이나 프로그램 등을 말한다.
- 프로그램이 정보를 적절히 조작할 수 있게 해 주는 데이터 구조 프로그램들의 사용과 운영을 기술하는 문서를 총칭한다.

❖ 특 징

- 소프트웨어는 개발되거나 설계되며 제조되는 것이 아니다.
- 소프트웨어는 조립하는 것이 아니라 사용자의 요구에 따라 새롭게 생산.
- 소프트웨어는 소모되지 않는다.
- 소프트웨어는 물리적 시스템요소가 아닌 논리적 시스템 요소이다.
- 비가시성 : 소프트웨어의 구조는 눈으로 볼 수 있는 것이 아니라, 내부 코드와 문서 형식으로 존재한다.
- 복잡성 : 프로그램 개발 과정에 여러 가지 상황이 복잡하며 얽혀 있으며, 누구나 시스템 자체를 이해하기가 어렵다.

❖ 품질

- 좋은 소프트웨어란?
 - 효율적이어야 하며 신뢰성이 높고 유지보수가 용이해야 한다.
 - 사용자수준에 따른 적당한 사용자 인터페이스를 제공해야 한다.
- 품질의 중요한 특성
 - 정확성 : 소프트웨어의 설계경향이나 코딩의 결함이 없는 정도
 - 신뢰성 : 지정된 시간동안 주어진 조건하에서 필요한 기능을 오류 없이 수행하는 능력
 - 강인성 : 프로그램 개발 당시에 구별되지 않은 상황이 발생하더라도 전체 동작에 영향을 주지 않는 성질
 - 성능(Performance), 사용의 용이성(User Friendliness), 유지보수성(Maintainability), 재사용성(Reusability), 이식성(Portability)

주요포인트 꼭 알고 가야해!

☞ 60년대에는 사용의 용이성보다는 효율성에 중점을 두었으나 최근에는 효율성보다는 사용의 용이성이나 유지보수성에 더 큰 비중을 둔다.

❖SDLC(Software Development Life Cycle)란?

- 정보시스템을 개발하는 절차, 혹은 개발단계의 반복현상을 시스템 개발주기(System Development Life Cycle)이라고 부른다.
- SDLC는 소프트웨어가 개발되기 위해 정의되고 사용이 완전히 끝나 폐기될 때까지의 전 과정을 단계로 나눈 것으로 소프트웨어의 분석, 설계, 구현, 유지보수를 각 공정 단계로서 구성한 모델이다.

❖소프트웨어 개발 단계

- 프로그램 개발의 시작부터 최종 개발 완료까지의 작업은 프로그램의 복잡도, 크기, 응용 프로그램의 영역에 관계없이 정의, 개발, 유지보수 과정을 수행하게 된다.
- 정의 단계
 - 프로그램 개발 시 어떤 프로그램을 개발 할 것인지를 정하는 것이다. 타당성 검토, 프로젝트 계획단계, 요구분석 단계로 구분된다.
- 개발 과정
 - 만들고자 하는 프로그램이 정해졌다면, 어떤 틀을 가지고, 어떤 방식으로, 어떤 데이터 구조를 사용해서, 사용자에게는 어떤 인터페이스를 제공할 것인지, 그리고 최종 테스트를 어떻게 해서 최종사용자에게 전달 할 것인지를 결정하는 단계이다. 설계 단계, 구현 단계, 검사 단계로 구분된다.
- 유지 보수 과정
 - 개발 시 정의된 규정에 의해서 프로그램이 완성이 된 후, 사용자가 추가 변경을 요청하거나, 미처 알지 못한 예외 및 기타 상황에 대해서 어떻게 대처를 할 것인지를 정하는 과정으로 가장 많은 시간과 비용이 투입되는 과정으로 전 과정의 약 60% 이상을 차지한다. 종류에는 수정, 적응, 기능 향상 그리고 예방 유지보수로 구분 될 수 있다.

❖소프트웨어 공학 패러다임(개발 주기)

- ① Waterfall Model
- ② Prototyping Model
- ③ 진화적 Prototyping Model
- ④ 4GT(4th generation technique) Model

주요포인트 꼭 알고 가야해!

주요포인트 꼭 알고 가야해!

GO

[04강] S/W생명주기모형

1. 폭포수 모형
2. 프로토타이핑 모형
3. 나선형과 4GT
4. IPT기법
5. HIPO
6. PDL, CPT, SW
7. N-S Chart, 검열
8. 모듈

합격
비법대한민국 대표 IT강사
조 대 호

여러분의 인생드라마는
지금 잘 진행되시나요?
그 드라마의 주인공은
바로 당신입니다.
이 세상 모든사람들은
당신을 위한 조연입니다.
당신의 드라마가
성공한 그날!
당신의 감사메시지 속에서
저의 이름이 불려지기를
기원합니다.

❖ 폭포수 모형(Waterfall Model)

- 1970년대에 소개되어 사용되던 모형으로 마치 물이 위에서 아래로 떨어지듯이 소프트웨어 개발 과정의 각 단계가 순차적으로 진행되고, 두 개 이상의 과정이 병행 수행되거나 이전 단계로 넘어가는 경우가 없는 모델
- 타당성 검토
 - 경영자가 소프트웨어의 필요성을 깨달아 이를 개발하기 위한 타당성을 검토
- 계획(Feasibility Study)
 - 사용자의 문제를 정의하고, 시스템이 갖추어야 할 기본 기능과 성능요건을 파악
- 요구분석(Requirement Analysis)
 - 사용자의 문제를 이해하고, 소프트웨어가 담당해야 하는 정보영역을 정의한다.
- 기본설계
 - 개발될 소프트웨어에 대한 전체적인 하드웨어 및 소프트웨어 구조, 제어구조, 자료구조의 개략적인 설계를 작성한다.
- 상세 설계
 - 제어구조, 자료구조, 인터페이스 구조, 크기, 주요 알고리즘, 각 프로그램 요소에 대한 알고리즘 등 프로그램 전 단계에 필요한 모든 규칙을 작성하는 단계이다.
- 코딩(Coding)
 - 각 모듈의 코딩과 디버깅이 이루어지며 결과를 검증하는 단위 검사(Unit Test)가 이루어진다.
- 테스트(Testing)
 - 통합 검사(Integration Test), 시스템 검사(System Test), 인수 검사(Acceptance Test)가 이루어진다.
- 실행
 - 완전한 기능을 갖춘 하드웨어 및 소프트웨어 시스템이 작동하는 단계이다
- 운용과 유지보수
 - 시스템이 고객에게 인도되어 설치된 후 실제업무에서 사용된다. 초기단계에서 발견되지 않는 오류를 수정하고, 새롭게 이식된 요구사항을 위해 시스템 구축과 서비스를 향상시키는 작업

주요포인트 꼭 알고 가야해!



2. 프로토타입 모형

❖ 프로토타입 모형(Prototyping Model)

- 프로토타입이란 개발자들과 사용자들의 의사소통상의 효과를 증진시키기 위하여 취하는 시스템개발상의 기법이다.

❖ 프로토타입이란?

- 프로그램 의뢰자가 최종 결과물이 만들어지기 전에 최종 결과물의 일부 혹은 모형을 보고자 할 때 일부 혹은 모형을 만드는 과정을 프로토타입이라 한다. 즉, 실제 상황이 나오기 전에 가상으로 시뮬레이션을 통해 최종 결과물에 대한 예측을 할 수가 있어 사용자가 쉽게 요구사항을 제시할 수 있고 치명적인 오류를 최소화한다.

❖ 폭포수모형과 프로토타입의 비교

	폭포수 모형	프로토타입 모형
장점 특징	응용 분야가 단순하거나 잘 알고 있는 경우에 적합하다. 각 단계가 끝난 후 결과물이 명확히 정의되어야만 한다. 개발 단계가 선형(Linear) 구조이다. 모든 분석은 프로젝트가 시작되기 전에 완성되어야만 한다.	사용자 요구사항의 정확한 파악 프로그램 의뢰자가 완성될 시스템의 모형을 볼 수 있다. 개발 단계에서 오류가 수정되므로 유지 보수가 필요 없다. 사용자의 요구를 확실히 알 수 있으므로 비용과 시간이 줄어든다.
단점	각 단계의 병렬 수행이 불가능하다. 오류 수정이 어렵고 개발 기간이 늦어진다. 문서화 중심의 프로세스가 많아 문서화에 치우치기가 쉽다. 사용자의 의견을 프로젝트 중간에 첨가시킬 수 없다.	발주자가 모형을 최종 결과라 믿는 오해가 발생한다. 발주자는 모형에 대한 기대심으로 개발 일정 단축을 요구하므로 소프트웨어 품질을 저하시킬 수 있다.

주요포인트 꼭 알고 가야해!



❖진화적 프로토타입 모델

- 최근에 가장 각광받고 있는 개발절차 모형으로 프로토타입을 지속적으로 발전시켜 최종 소프트웨어 개발까지 이르는 개발방법으로 위험관리가 중심인 생명주기 모델이다. 위험분석에 대한 해결역량이 없으면 오히려 위험한 모델이 될 수 있다
- 작업 단계별 기능
 - 계획 수립(Planning) : 목표, 다른 방안, 제약 조건을 결정.
 - 위험 분석(Risk Analysis) : 위험요소 및 타당성을 분석하여 관리 기술을 통해 해결.
 - 개발(Engineering) : 개선된 한 단계 높은 수준의 제품을 개발
 - 고객 평가(Customer Evaluation) : 개발된 프로토타입을 고객이 평가.
- 특징
 - 대규모 시스템의 소프트웨어 개발하는데 적합
 - 개발과 유지보수 사이의 구분이 없다.
 - 소프트웨어의 재사용이 가능하다.
 - 프로젝트의 예상되는 실패 원인을 사전에 막을 수 있다.
 - 초기에 위험 요소를 발견하지 못할 경우 위험 요소를 제거하기 위해서 많은 비용이 들 수 있다.

주요포인트 꼭 알고 가야해!



❖4GT(Fourth Generation Language)기법

- 4세대 기법은 하나의 일을 공통적으로 수행하는 소프트웨어 도구들의 광범위한 배열을 포함한다. 소규모 애플리케이션에서 비절차 4세대 언어(4GL : Fourth Generation Language)를 이용해 요구 사항 수집을 구현으로 직접 전환시킬 수 있다.
- 4세대 기법의 개발 단계
 - 요구수집 → Design → 4GL을 이용한 생성 → Testing
- 특징
 - 자연언어(4GL 사용)로 원시 코드 자동 생성한다.
 - 설계단계축소
 - 중소 규모의 프로젝트 개발 시 시간 단축이 가능하다.

4. IPT 기법

❖IPT란?

- IPT는 소프트웨어 공학에서 사용하는 용어로 프로그래밍 개선기법 또는 효과적 프로그램 개발 기법이라고도 한다. 소프트웨어 개발 단계에서 여러 가지 표준화된 도구를 사용하여 생산성을 높이는 방법이다.
- IPT는 HIPO, N-S chart, PDL(pseudo code) 등의 도구와 구조적 기법, top down programming, structured coding, chief programmer, walkthrough 등을 사용하는 기법이다.

❖분류

- 기술면
 - 설계 분야에서의 복합 설계(Composite Design)
 - 코딩 분야에서의 구조적 코딩(Structured Coding)
 - 테스트 분야에서의 하향식 프로그래밍(Top-Down Programming)
 - 설계 보조 수단으로서의 HIPO(Hierarchy plus Input Process Output)
 - 프로그램 기술 언어(Program Description Language)
 - 나씨-슈나이더만 차트(Nassi-Schneideman Chart)
- 관리면
 - 개발 조직으로서의 선임 프로그래머 팀(Chief Programmer Team)
 - 품질 관리상의 워크스루(Walk-Through)와 검열(Inspection)
 - 개발 지원 라이브러리(Library)

❖구조적 프로그래밍

- 순차(sequence), 반복(iteration), 선택(selection)구조에 의해 규칙적이고 논리적인 블록 단위의 프로그램 기법
- 단일 입구, 단일 출구 구조
- 무조건 분기문(GO TO)의 사용을 배제

주요포인트 꼭 알고 가야해!

[하향식 프로그래밍(Top Down Programming)]

프로그램 코딩과 테스트 등을 상위 모듈로부터 하위 모듈로 진행해 나가는 기법이며 문제를 분해하여 세분화된 내용으로 관련 요소를 정의

[구조적 설계(Structured Design)]

계층적 구조에 맞추어 프로그램의 위 단계부터 아래 단계로 처리해 나가도록 설계하는 기법으로 모듈단위로 분할하여 설계하고 인터페이스를 명확히 표현한다.

❖HIPO의 정의

- HIPO는 Hierarchy Plus Input Process Output의 약자로서 시스템의 설계나 또는 시스템 문서화(Documentation)용으로 사용되고 있는 기법으로 개요 설계에서 사용된다. 하향식 개발에 용이하고 개발과정에서 문서화를 부산물로 얻을 수 있으며 목적에 알맞은 수준의 자료를 볼 수 있다. 그러나 조건, 선택, 반복 구조 등을 표시할 수 없고 자료 구조 또는 자료간의 관계를 표시할 수 없다.

❖HIPO 모형의 종류

- 도식 목차(Visual Table of Contents)
 - 일련의 HIPO 문서의 구성을 나타내는 계층 구조도이다. 다이어그램의 구조와 각 기능의 관계를 도식화한 것으로, HIPO에서 지정된 기능을 계층적으로 나타내는 것이다.
- 총괄 다이어그램(Overview Diagram)
 - 프로그램을 구성하는 기능을 기술한 것으로 입력, 처리, 출력에 대한 전반적인 정보를 제공한다.
- 상세 다이어그램(Detail Diagram)
 - 총괄 다이어그램에 의해 나타난 기능을 구성하는 기본적인 요소를 기재하는 것으로, 특정의 입력/출력 항목을 기술한다.

❖HIPO의 이점

- 하향식(Top-Down) 개발이 쉽다.
- 문서의 체계화가 가능하다.
- 보기 쉽고 알기 쉽다.
- 기능과 자료의 의존 관계를 동시에 표현할 수 있다.
- 변경, 유지 보수가 용이하다.
- 구상에 도움을 준다.

주요포인트 꼭 알고 가야해!

❖프로그램 기술 언어(PDL) = 의사 코드(Pseudo Code)

- 모듈의 논리를 설계할 때 사고 과정을 체계화해 가는 방법론의 하나이며, 하향식 접근 방식으로 논리의 전체 흐름을 최초로 표현하게 된다.
- 이점
 - 특정한 문법이나 언어의 제약이 없다.
 - 프로그래밍 언어를 모르는 사람도 작성이 가능하다.
 - 프로그램의 상세 설계용으로 사용 될 수 있다.
 - 프로그램 언어 교육을 위한 수단으로 사용할 수 있다.
 - 문서의 목적 또는 시스템 재검토 목적으로 사용할 수 있다.
 - 흐름도에 비해 내용 전달이 용이하다.
- 유의사항
 - 기능의 범위, 즉, 단위 기능을 파악할 수 있도록 한다.
 - 계층적 표현이나 네스팅에 의한 구조 표현 방식을 사용하여 문장의 의미나 또는 그 기능 등을 명확히 한다.
 - GOTO 등의 표현은 피한다.

❖선임 프로그래머 팀(Chief Programmer Team)

- 능력과 경험이 풍부한 책임 프로그래머를 중심으로 하여 팀을 구성하고, 이와 같은 팀들을 바탕으로 프로그램을 개발하는 방법
- 책임 프로그래머는 개발 팀의 최고 책임자로서 해당 팀에 할당된 소프트웨어의 주요 부분을 제작하며, 팀의 활동을 조정하며, 팀의 다른 구성원들의 작업 내용을 검토하고 개발 중인 프로그램의 기술적인 지원을 한다.

❖구조적 검토회(Structured Walkthrough)

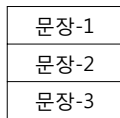
- 시스템 명세, 설계 결과, 프로그램 코드 등을 각각 여러 사람이 검토하게 함으로써 그 안에 포함되어 있을 수 있는 오류를 조기 발견하고자 하는 것이 검토회이다. 오류는 그것이 발생한 후 시간이 경과함에 따라 수정하기 위한 비용은 점차로 증대된다. 조기 발견이 비용절약에 주는 효과는 크다.

주요포인트 꼭 알고 가야해!

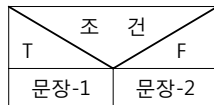
❖나씨-슈나이더만 도표(N-S Chart, Box Diagram)

- N-S 차트는 분석설계 도구로는 부적합하고 상세 설계 표현 도구로 쓰이며 하나의 입구와 출구가 있는 프로그램의 구조를 나타내기엔 편리하다.
- N-S 차트는 논리적인 구조를 4각형의 박스로 표시하는데 연속, 선택, 반복 등의 구조를 사용하고 분기명령은 허용하지 않는다.

① 순차 구조



② 선택 구조



③ 반복구조



❖검열(Inspection)

- 검열은 검토회와 유사하다. 다만, 검토회가 발견한 오류를 문서화하고, 이를 개발자 자신의 오류 수정에만 활용하는 데 반하여 검열은 발견된 오류를 문서화할 뿐만 아니라 이것을 관리용 자료로서 사용한다. 즉, 오류는 형태별로 분류하여 그 빈도를 집계하고 정리한다. 이것들은 다음에 있을 검열에서 오류 발견을 용이하게 하기 위하여 이용된다. 다시 말하면, 검토회를 보다 더 시스템화하여 그 결과를 버리지 않고 정리해서 경험 자료로 만들고 이를 축적하여 훗일에 반영할 수 있도록 하는 것이 검열이다.

❖개발 지원 라이브러리

- 개발 환경을 정비함으로써 관리를 용이하게 하고, 그 결과로서 품질과 생산성의 향상을 꾀하고자 하는 기법이다. 컴퓨터로 관리할 수 있는 것은 전부 내부 라이브러리로 해서 컴퓨터로 관리하고, 그 복사 문서는 외부 라이브러리에서 관리한다. 외부 라이브러리는 시스템 개발을 위한 참가자들이 공동으로 정보 교환을 하는 장소로서 활용한다.

주요포인트 꼭 알고 가야해!

❖모듈 설계(Module Design)

- 모듈이란 소프트웨어의 구성 요소로서 서브루틴, 단위 프로그램, 단위별 작업 처리 내용, 동일한 작업 처리 내용을 담고 있는 서브시스템 등을 의미한다.

❖모듈 설계 지침

- 소프트웨어 요소간의 효과적 제어를 위해 설계에서 계층적 자료조직이 제시되어야 함
- 소프트웨어는 논리적으로 특별한 기능과 부기능을 수행하는 요소들로 나누어져야 함
- 설계는 자료와 프로시저에 대한 분명하고 분리된 표현을 포함해야 한다.
- 모듈의 작성은 분담하여 독립적으로 작성할 수 있다.
- 모듈은 서로 결합되어 종속적으로 실행된다.
- 컴파일만족은 독립적이다.
- 모듈은 업무 성격이 비슷한 처리에 부품처럼 공통으로 사용할 수 있다.
- 모듈은 이름을 정의하여 필요시 사용되고 전달되는 매개 변수(인수)를 제공한다.

❖모듈 작성 시 주의 사항

- 적절한 크기로 작성되어야 한다.(size)
- 모듈의 내용이 다른 곳에 적용 가능하도록 표준화하여야 한다. (reusability)
- 모듈간의 기능적 결합도는 적게 한다. (coupling의 최소화)
- 모듈 내의 요소들끼리의 응집도는 크게 한다. (cohesion의 최대화)
- 자료의 추상화와 정보 은닉의 성격을 띄도록 해야 한다. (abstract data type, information hiding)
- 보기 좋고 이해하기 쉽게 작성해야 한다.

❖Fan-out

- 소프트웨어 구조도 내에서 어느 한 모듈이 직접 제어할 수 있는 모듈의 수

❖Fan-in

- 소프트웨어 구조도 내에서 어느 한 모듈을 제어하는 모듈의 수

주요포인트 꼭 알고 가야해!

Ex>아래의 그림에서 F모듈의 Fan-in과 Fan-Out 은 얼마인가?



주요포인트 꼭 알고 가야해!

(GO)

[05강] 구조적설계방법론

1. 구조적 시스템
2. 요구분석
3. 요구분석을 위한 자동화기법
4. 자료흐름도
5. 자료사전과 미니명세서
6. 구조적설계의 원리
7. 기타 설계 방법론

합격
비법대한민국 대표 IT강사
조 대 호

여러분의 인생드라마는
지금 잘 진행되시나요?
그 드라마의 주인공은
바로 당신입니다.
이 세상 모든사람들은
당신을 위한 조연이랍니다.
당신의 드라마가
성공한 그날!
당신의 감사메시지 속에서
저의 이름이 불려지기를
기원합니다.

정보처리기사/산업기사

합격은 당연! 고득점 합격이 목표!

(GO)

1. 구조적 시스템

JODAEHO

gisa plus

❖구조적 시스템의 특성

- 소프트웨어의 품질이 일정치 못하고 예산 및 일정의 초과를 가져오는 등 하드웨어 생산의 발전 속도에 비해 소프트웨어 생산 분야에서는 오래전부터 '소프트웨어 위기'를 겪어왔다. 프로그램 언어 지식을 갖춘 개발자들이 그들 나름대로 필요한 모듈들을 작성하고 조합해가면서 원칙 없는 개발에 매달렸지만 프로그램의 본 수가 늘어갈수록 많은 불량률(에러율)과 비구조성으로 유지보수에도 많은 어려움을 유발했다.
- 60년대 후반에는 이른바 'GOTO 논쟁' 즉, 프로그램을 짜는데 있어서 GOTO문은 판독성을 해치고 프로그램의 구조를 복잡하게 만들므로 쓰지 말아야 한다는 주장이 제기되었다. GOTO 분기 대신에 '순차(Sequencing), 선택(Selection), 반복(Iteration)'으로 모든 로직이 해결가능하며 이해하기가 쉬워 좋은 프로그램이 된다는 것이다. 이것이 바로 구조적 프로그래밍의 원리이다.
- 그러나, 연구가 거듭되면서 소프트웨어의 개발에 있어서 구조적인 프로그래밍 보다는 분석, 설계 등 그 이전 과정에서도 구조적인 접근방법이 동원되어야 소프트웨어 개발의 생산성과 품질을 높일 수 있다는 이론이 설득력을 얻게 되었는데 이것이 바로 구조적 분석과 설계이며 이 모든 것을 종합하여 구조적 방법론이라고 일컫는다.

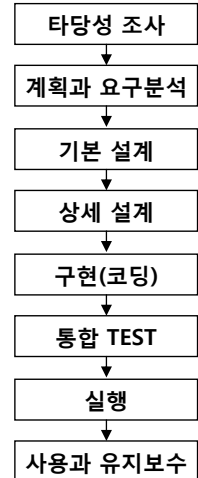
❖구조적 시스템의 원리

- 구조적 분석과 설계는 구조적 프로그래밍에서 출발한 만큼 구조적인 프로그래밍이 이루어질 수 있는 분석과 설계가 되도록 유도하고 있다. 가장 큰 특징은 두서없는 프로그래밍이 되지 못하도록 미리 전체 기능의 관점으로부터 Top-Down의 형태로 진행되는 점이며, 철저하게 모듈화하여 분석, 설계, 프로그래밍이 이어진다는 점이다.

❖요구분석

- 요구사항 분석은 사용자의 요구사항을 명확히 규정하고 시스템의 특성을 반영하는 과정이며, 이 단계에서 사용자의 뜻을 이해하고 업무를 분석한다. 요구사항 분석은 사용자의 막연한 문제의식이나 요구로부터 시스템이나 소프트웨어의 목적, 수행할 작업들을 요구조건으로 분명하게 명문화하는 과정이며, 요구 조건을 만족하도록 논리적으로 요구 명세서로 기술하는 것도 포함한다.

주요포인트 꼭 알고 가야해!



정보처리기사/산업기사

합격은 당연! 고득점 합격이 목표!

(GO)

[05강] 구조적설계방법론

Slides 32

❖요구 분석의 단계

- 문제 인식
 - 개발 과정과 관련된 개체, 관계, 기능, 프로세스, 제약 등의 중요한 사항들을 판단하는 작업
- 전개
 - 이미 파악한 문제에 대해서 더 자세하게 조사하는 것으로서, 사용자가 어떻게 자료에 대하여 질문하며, 어떤 방식으로 자료를 조사하기를 원하는지, 또한 어떤 식으로 결과물을 출력되기를 원하는지, 자료는 어디에 위치하기를 원하는지 판단하는 과정.
- 종합
 - 다이어그램 형태로 표현하며 자동화 도구를 사용하여 통합 모형으로 표현한다.
- 검토
 - 결과, 일정, 조직 등을 검토하고 재정비한다.
- 문서화
 - 개발 과정 및 시스템의 여러 상황(기능, 성능, 정보 표현, 제약, 검증 평가 기준)에 대하여 설명

주요포인트 꼭 알고 가야해!

❖요구 분석 명세서

- 소프트웨어가 가져야 하는 기능과 능력을 명시한 것으로 주어진 입력에 대하여 시스템이 어떻게 동작해야 하는가를 명시하며 자료 흐름도에서 나타낼 수 없는 시스템에 관련된 중요한 사항을 나타낸다.
- 작성법
 - 자료 흐름도, 자료 사전, 소단위 명세서로 표현된 현재 및 제안 시스템의 모형에 분석가가 제약 조건 및 인수 기준을 첨가해서 완성하는 것이다.
- 평가 기준
 - 무결성과 완벽성, 일관성과 명확성, 명확성과 추적가능성, 기능적, 검증가능성

3. 요구분석을 위한 자동화기법

❖SADT(Structured Analysis and Design Technique)

- 시스템 정의, 소프트웨어 요구사항 분석, 그리고 시스템/소프트웨어 설계를 위해 널리 사용되고 있는 시스템 분석과 설계기법이다.
- 분석가에게 소프트웨어 기능을 분해하도록 해주는 프로시저들로 구성된다. 즉 그래픽 표기법, 소프트웨어 내에 정보와 기능의 관계와 방법에 적용하기 위한 프로젝트 제어 지침을 의사 교환해주는 SADT활동도와 자료도 등이다.

❖SREM(Software Requirement Engineering Methodology)

- 요소, 속성, 관계, 구조들을 기술하기 위하여 RSL(요구사항을 기술하기 위한 언어)을 사용해서 만든 자동화 요구사항 분석도구이다.

❖PSL/PSA(Problem Statement Language/Analyzer)

- 미시간 대학의 ISDOS 프로젝트에서 개발된 요구 분석용 자동화 도구
- 요구분석에 필요한 내용을 PSL이란 기술언어를 사용하여 작성 한 뒤 PSA는 분석 데이터베이스에 저장하고 있는 자료를 분석하여 최적의 요구명세서를 자동으로 출력해준다.

❖TAGS(Technology Automated Generation of System)

- 통합 CASE 도구, 소프트웨어 개발 및 유지보수의 전 과정에 이용

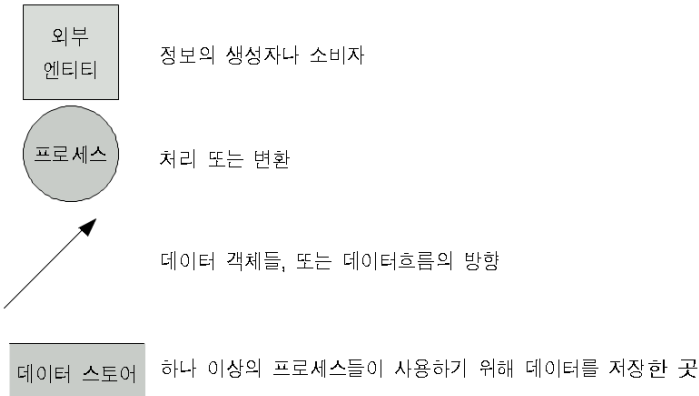
주요포인트 꼭 알고 가야해!

❖구조적 분석 절차

- Data Flow Modeling : 시스템 자료의 흐름을 모형화한다.(DFD 이용)
- Data Modeling : 시스템에 필요한 자료를 모형화한다.(DD, E-R Diagram 이용)
- Data Process Modeling : 자료가 어떻게 처리되는지 자료 이용 방법과 자료 처리 방법을 모형화한다.(의사 결정표, 의사 결정 트리, 구조적 언어 이용)

❖자료 흐름도(DFD : Data Flow Diagram = Bubble Chart)

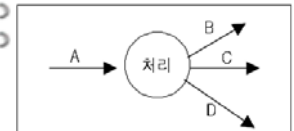
- 자료가 소프트웨어의 각 절차를 따라 흐르는 과정을 모형화 하는데 사용되는 도구
- 설계의 기본이 되며 현재 시스템의 자료 흐름 분석에 사용
- 데이터 흐름을 변형시키는 기능을 묘사한다.
- 배경도로 시작된 DFD는 버블을 점차 분해, 상세화 시켜가면서 세부 모양을 표현한다.
- 표기법



주요포인트 꼭 알고 가야해!

[정보의 흐름 유형]
· 변환중심

· 거래중심



5. 자료사전과 미니명세서

❖자료 사전(DD : Data Dictionary)

- 자료 흐름도의 대상이 되는 자료의 내용을 구체적으로 명시한 것
- 자료 항목, 속성 등이 정의되며 보통 DFD에서 사용되는 모든 자료를 규정하는데 사용되는 도구

기 호	의 미
=	"is composed of"
+	"and", "along with"
{ }	"iteration of"
[]	"alternation"
()	"optional"
**	"comment"

❖미니 명세서(Mini Spec)

- DFD의 최하위 버블에서 어떤 일이 처리되는지 로직(Logic)을 개략적으로 기술하는 산출물이 소단위 명세서, 즉 미니 스펙이다.
- 소단위 명세서로 가장 많이 이용되는 도구로는 구조적 언어(Structured Language), 의사 결정도(Decision Tree), 의사 결정표(Decision Table), N-S 도표(N/S Chart) 등이 있다.

❖소단위 명세서의 표기법

- 구조적 언어 형식
 - 한정된 단어 : DD에 정의된 자료만을 이용
 - 한정된 문형 : 간단한 명령문과 대수 방정식만 사용
 - 제한된 구조 : 순차, 선택, 반복 구조만을 이용
- 의사 결정표(Decision Table) : 의사 결정의 요소가 많을 때 사용하고 의사 결정 변수들이 취할 수 있는 값이 여러 개일 때와 조건의 완전성을 판단하기 어려울 때 사용
- 의사 결정 나무(Decision Tree) : 의사 결정 나무는 의사 결정도를 분기 구조로 나타낸 것이다.

주요포인트 꼭 알고 가야해!

$$A = \{ B + C \}^2$$

[소단위 명세서의 역할]

- DFD에서 설명되지 않은 처리 과정 자체를 설명
- 입력 자료를 출력 자료로 변환시키기 위한 업무 내용 기술
- DFD의 최하위 버블(기능 요소) 각각에 대하여 하나의 미니 명세서로 작성

❖구조 설계(개요 설계)

- 소프트웨어의 내부 구조와 자료 구조를 결정하는 단계이다. 구조적 설계의 결과로는 시스템 구조도, 프로그래밍 사양서, 자료 설계서가 만들어지게 되고, 시스템 구조도는 소프트웨어를 구성하는 모듈의 정의, 모듈 사이의 관계를 나타내는 문서가 만들어지게 된다.

❖설계 원리

- 자료 흐름도의 프로세스 형식이 시스템의 구조를 결정하며, 분할과 정복. 즉, 큰일을 작게 쪼개고, 쪼갠 작은 일을 하나씩 풀어나가는 과정을 계속 반복하는 것으로 아래에서 보는 것과 같이 크게 입력, 프로세스, 출력 부분으로 나누어진다.
- 추상화(Abstraction)
 - 문제를 이해하고 표현하기 위해 세부사항을 모두 기술하지 않고도 추상화, 또는 개념화시켜 표현하는 원리이다.
 - 기능 추상화, 자료 추상화, 제어 추상화의 3종류로 구분
- 정보 은닉(Information Hiding)
 - 각 모듈은 다른 모듈에 독립적이며 한쪽 모듈의 변경이 다른 모듈의 세부내용에 영향을 미치지 않는다.
- 모듈화(Modulization)
 - 하나의 시스템을 서브시스템, 프로그램, 모듈 등으로 구분하여 정의하고 개별적으로 설계한다. 모듈이란 이름을 가지며 하나의 작업단위를 처리할 수 있는 최소한의 프로그램 단위라고 볼 수 있다
- 구조화(Structuring, 분할과 정복)
 - 소프트웨어에 계층적인 구조를 부여하여 상위 모듈이 하위 모듈을 활용하는 질서를 갖도록 한다.
- 단계적 정제화
 - 하향식으로 진행하면서 점차적으로 내용을 구체화시켜가는 것이다.

주요포인트 꼭 알고 가야해!

[구조 설계의 작업 순서]

- ① 자료 흐름도로부터 시스템 구조도를 만든다.
- ② 구조도를 수정해서 완성본을 만든다.
- ③ 사용하게 될 파일 구조 및 데이터베이스를 설계한다.
- ④ 프로그램 사양서를 만든다.
- ⑤ 사용자에게 보여 질 인터페이스를 구성한다.

7. 기타 설계방법론

❖Dijkstra(다익스트라) 설계 방법

- GOTO 문의 배제, 저급한 제어 구조의 배제, 프로그램의 구조 복잡화의 회피
- 프로그램의 동적 처리와 정적 표현의 격차를 좁혀 구조를 분명하게 하자는 주장

❖Yourdon(요든) 설계 방법

- DFD(Data Flow Diagram) 방법론을 제안
- 순차, 선택, 반복 구조를 사용하고 데이터 흐름 중심의 처리 주장
- 외부 입출력, 처리 과정, 데이터 흐름 방향, 데이터 저장소로 표현

❖Jackson(잭슨) 설계 방법

- 프로그램이 다루는 입출력 데이터의 구조를 설계하여 프로그램의 제어 구조를 유도하는 방법
- 기본, 순차, 선택, 반복의 4가지 구조를 사용
- 입력 데이터 구조와 출력 데이터 구조에 관한 설계 내용으로 대응 관계형성

❖Warnier-Orr(워니어-오어) 설계 방법

- HIPO 또는 구조화 도표에 대한 대체 방법으로 데이터 구조적 시스템 개발에 사용
- 선택과 반복 구조만으로 프로그램을 표현
- 시스템의 출력을 근거로 데이터 구조를 결정하고 이를 기반으로 프로그램 구조 결정

❖상세 설계

- 프로그램의 각 모듈 안에서 발생하는 일들과 사용자 인터페이스에 대하여 계획한다. 각 모듈 안에서 사용하게 될 알고리즘을 생각하고 자료 구조에 관해서도 계획한다. 또한 사용자가 시스템에 어떻게 접근하도록 할 것인가를 설계하는 인터페이스 설계도 해야 된다.

주요포인트 꼭 알고 가야해!

[상세설계의 단계]

- ① 프로그램 설계
- ② 자료구조 설계
- ③ 인터페이스 설계
- ④ 상세 설계서

주요포인트 꼭 알고 가야해!

(GO)

[06강] 구조적개발방법론

1. 결합도
2. 응집도
3. 구현
4. 검사
5. 통합과 테스트
6. 유지보수
7. 형상관리

합격
비법

대한민국 대표 IT강사
조 대 호

여러분의 인생드라마는
지금 잘 진행되시나요?
그 드라마의 주인공은
바로 당신입니다.
이 세상 모든사람들은
당신을 위한 조연입니다.
당신의 드라마가
성공한 그날!
당신의 감사메시지 속에서
저의 이름이 불려지기를
기원합니다.

정보처리기사/산업기사

합격은 당연! 고득점 합격이 목표!

(GO)

1. 결합도

JODAEHO

gisa plus

❖ **내용 결합도(content coupling)**

- 한 모듈과 다른 모듈이 그 내용의 일부를 공유하는 것으로, 한 모듈이 다른 모듈 내의 명령이나 내부 데이터 값을 참조하거나 변경할 때 두 모듈간의 관계를 말한다.

❖ **공통 결합도(common coupling)**

- 공유 영역에 정의한 데이터를 몇 개의 모듈이 공통으로 사용할 때의 관계를 말한다.

❖ **외부 결합도(external coupling)**

- 외부 선언된 자료를 공유하는 결합이다. 자료를 공유한다는 점에서는 공통 결합도와 같지만 외부 선언된 자료만을 공유하므로 공통 결합도에서와 같이 불필요한 자료는 공유하지 않는다.

❖ **제어 결합도(control coupling)**

- 한 모듈이 다른 모듈 내의 기능을 제어하기 위해 서로 제어용 신호를 주고받는 관계를 말한다.

❖ **스탬프 결합도(stamp coupling = 형태 결합도)**

- 두 개의 모듈이 일부분의 동일한 자료 구조를 참조하는 관계로 모듈간의 인터페이스로 배열이나 레코드 등의 자료구조가 전달된 경우를 말한다.

❖ **자료 결합도(Data coupling)**

- 모듈간의 인터페이스가 자료요소로만 구성된 경우로 모듈 간에 필요한 자료만 주고받는 관계를 말한다.
- 단일 파일 또는 동종 테이블을 매개변수로 통신하는 모듈간의 결합성을 의미한다.

주요포인트 꼭 알고 가야해!

[구조적 설계 평가 기준]

구조적 설계의 평가는 모듈화(모듈간의 독립성, 모듈의 기능)가 얼마나 잘 되어 있는가를 가지고 평가하며, 평가의 척도로 결합도와 응집도를 사용한다.

[결합도의 정의]

결합도는 두 모듈 간의 상호 의존도를 측정하는 것이다. 이는 모듈간의 연관성을 의미하는 것으로 좋은 설계가 이루어지도록 하기 위해서는 가능한 한 모듈을 독립적으로 생성함으로써 결합도를 최소화(낮게)하는 것이다.

정보처리기사/산업기사

합격은 당연! 고득점 합격이 목표!

(GO)

[06강] 구조적개발 방법론

Slides 40

❖기능적 응집도(functional cohesion, 함수응집도)

- 모듈 내부의 모든 기능 요소들이 단일 문제와 연관되어 수행하는 경우를 말한다. 응집도 중에서 가장 강하고 가장 좋은 모듈이다.

❖순차적 응집도(sequential cohesion)

- 모듈의 구성 요소가 하나의 활동으로부터 나온 출력 자료를 그 다음 활동의 입력 자료로 사용하는 같은 모듈 내에서의 응집의 정도를 나타내는 것이다.

❖정보적 응집도(communucational cohesion, 통신적응집도)

- 한 모듈에 2개 이상의 기능적 요소가 존재하면 동일한 입력 자료를 사용함에도 불구하고 서로 다른 출력자료를 생성하는 요소로 구성되며, 각 요소 사이에서 데이터를 주고 받거나 데이터를 공유하는 관계를 말한다.

❖절차적 응집도(procedural cohesion)

- 문제를 처리하기 위해 관계되는 여러 개의 기능 중에서 몇 개를 실행할 때 이 기능들이 순서대로 실행되는 관계를 말한다.(제어는 한 구성요소로부터 다른 구성요소로 흐르지 만 자료는 한 구성요소로부터 다른 요소로 흐르지 않는다.)

❖시간적 응집도(temporal cohesion)

- 여러 개의 순차적인 기능을 수행하지만 실행하는 기능 사이에 강한 관련성은 없으며, 시간적으로 수행시기가 같은 기능끼리 존재한다.

❖논리적 응집도(logical cohesion)

- 어떤 기능을 논리적이고 추상적으로 취하여 모듈화한 경우를 말한다. 즉, 유사한 성격을 갖거나 특정형태로 분류되는 처리요소들로 하나의 모듈이 형성되는 경우이다.

주요포인트 꼭 알고 가야해!

[응집도의 정의]

응집도는 한 모듈 내 구성 요소들의 기능적 관련성을 나타내는 척도로서 명령어의 모임, 호출문, 특정 작업 수행 코드 등 한 모듈 내부의 처리 요소들 간의 기능적 관계를 말한다. 즉 모듈을 이루고 있는 각 요소들이 공통의 목적을 달성하기 위하여 얼마나 관련이 있는가를 나타내는 것이다.

❖우연적 응집도

(coincidental cohesion)

- 모듈 내의 요소들 간에 특별한 관계가 없는 경우를 말한다. 모듈 내의 각 요소가 다른 모듈 내의 요소와 강한 연관성이 있어 내용이 변경되면 많은 영향을 받는다.

❖결합도

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

❖응집도

- 기능적 응집도 > 순차적 응집도 > 정보적 응집도 > 절차적 응집도 > 시간적 응집도 > 논리적 응집도 > 우연적 응집도

❖구현

- 프로그래밍 또는 코딩이라고 불리며 설계명세서가 컴퓨터가 알 수 있는 모습으로 변환되는 과정이다. 코딩 단계에서 결정해야 할 사항은 프로그래밍 언어의 선택과 프로그래밍 스타일에 대한 표준을 확립하고, 모듈을 어떤 순서대로 구현할 것인가 하는 것.

❖전통적 접근법

- 각각의 시스템은 따로 따로 작성되고, 테스트하며, 디버깅한다.
- 시스템의 모든 단위를 디버깅한 후, 모든 단위를 서브시스템으로 구성하고, 각각의 서브시스템을 테스트하며, 디버깅한다.
- 완전한 시스템을 형성하기 위해서 서브시스템을 결합, 테스트하고 디버깅.

❖점진적 접근법

- 하향식 접근법 : 스택이라고 부르는 모형 모듈을 갖는 최상위 모듈을 가장 먼저 수행하고 점차적으로 하위 모듈 순으로 수행해 나가는 기법
- 상향식 접근법 : 가장 하단 모듈로부터 시작하여 점점 상단 모듈을 구현해 가는 방식.
- 절충식 접근법 : 상향식 구현 방식과 하향식 구현 방식을 조합한 방식으로 시스템의 하단 반쪽을 상향식으로 수행하여 중간에서 만난다.

주요포인트 꼭 알고 가야해!

❖가장 좋은 품질은 결합도는 낮고 응집도는 높아야 한다.

❖검사의 정의

- 시스템이 결과물이 정해진 규칙대로 작동하는지, 예상과 실제 결과가 어떤 차이를 보이는지, 수동 또는 자동 방법을 동원하여 검사하고 평가하는 모든 과정.

❖소프트웨어 검사 단계

- 단위검사(코드검사)
 - 구현단계에서 원시 프로그램의 모듈단위로 테스트가 이루어진다.
- 통합검사(설계검사)
 - 단위 검사를 통과한 단위모듈로부터 프로그램의 구조를 건설하는 체계적 작업.
 - 인터페이스와 관련된 오류들을 발견하고 제거하면서 모듈들을 체계적으로 조합시키는 작업.
- 정당성검사(확인 or 검증 Test)
 - 사용자의 모든 요구사항이 시스템에서 완벽하게 수행하는지를 검사한다.
- 시스템검사
 - 소프트웨어와 다른 시스템 요소들을 전체적으로 검사하고 전체 시스템의 실행 능력을 확인
- 인수검사
 - 최종 만들고자 하는 시스템이 사용할 수 있도록 모든 준비가 되어 있는지를 검사하는 것으로 개발자가 하는 것이 아니라 개발 의뢰자나 그 대리인이 한다.
 - 알파 테스트 - 선택된 사용자가 개발 환경에서 시험하는 것으로 고객이 개발자의 위치에서 개발자와 함께 수행된다.
 - 베타 테스트 - 고객의 사용 환경에서 시험하는 것으로 개발자 없이 사용 환경에 소프트웨어를 설치하여 시험한다.
- 설치검사
 - 실제 사용될 환경에서 설치한 후 제대로 작동되는지 검사한다.

주요포인트 꼭 알고 가야해!

5. 통합과 테스트

❖하향식 통합

- ① 전체프로그램을 매번 실행하고 종속적인 모듈은 가짜 모듈(stub)로 대치
- ② 통합방법에 따라 가짜 모듈을 실제 모듈로 대치
- ③ 각 모듈을 통합되면 검사를 실시
- ④ 통합될 때마다 회귀 검사를 실시

❖상향식 통합

- ① 낮은 수준의 모듈들을 클러스터로 결합
- ② 드라이버라는 제어 프로그램의 작성
- ③ 클러스터의 검사
- ④ 드라이버는 제거하고 클러스터를 상위로 결합

❖화이트 박스 테스트

- 원시 프로그램을 직접 보며 모든 논리적인 경로를 직접 검사하는 방법이다.
- 세부적 오류, 논리 구조상의 오류, 반복문의 오류, 수행 경로 오류를 찾을 수 있다.
- 기초경로검사(Structure testing), 루프검사(Loop testing), 데이터흐름검사(data flow testing) 등의 종류가 있다.

❖블랙박스 테스트

- 프로그램을 실행시켜 데이터를 입력한 결과로 검사하는 방법으로 기능 테스트라 한다.
- 부정확하거나 빠진 오류, 인터페이스 오류, 자료 구조상의 오류, 성능 오류, 시작과 종결 상의 오류를 찾을 수 있다.
- 동치 분할 검사(equivalence partitioning testing), 원인 효과 그래픽 기법(cause effect graphic-technique), 비교 검사(comparison testing), 한계 값 분석(Boundary value analysis), 오류 예측(Error guessing) 등의 종류가 있다.

주요포인트 꼭 알고 가야해!

- 스터브 : 가짜 모듈
- 클러스터 : 관련된 하위 모듈의 그룹
- 드라이버 : 임시적으로 클러스터를 작동하여 검사하도록 사용하는 프로그램

❖유지보수

- 최종 소프트웨어를 개발 의뢰자에게 양도 후 실제 환경에서 소프트웨어를 사용하는 동안 오류가 발견되면 수정되어야 하고, 새로운 기능을 요청 시 추가해야 한다. 소프트웨어를 개발하는 기간보다는 유지보수 기간이 더 중요하며 시간과 비용이 더 많이 소모된다.

❖유지보수의 비용

- 소프트웨어 개발에 사용되는 전체 비용 중에 유지 보수에 사용되는 비용이 60% 정도로 많은 비용이 들어간다. 소프트웨어의 특성이 계속 개선되는 것인지 현재에 고정적으로 유지되는지는 않는다. 따라서 유지 보수를 쉽게 할 수 있도록 만드는 일이 상당히 중요하다.

❖유지보수의 형태

- 수정 유지보수(Corrective maintenance)
 - 소프트웨어 버그의 원인을 찾아 정상적으로 가동되도록 하는 유지보수작업이다.
- 적응 유지보수(Adaptive maintenance)
 - 새로운 데이터 파일이나 운영체제에서 적응될 수 있도록 기존의 소프트웨어를 변경하는 유지보수이다.
- 완전 유지보수(Perfective maintenance, 기능향상 유지보수)
 - 사용자는 소프트웨어를 사용하면서 소프트웨어에 대한 새로운 성능을 요구하게 되고, 사용하고 있는 기능을 수정하도록 요구될 때의 유지보수로 전체에서 50%를 차지한다.
- 예방 유지보수(Preventive maintenance, 재공학)
 - 미래의 유지보수성이나 신뢰성을 개선시키기 위해서, 또는 미래의 성능향상을 위한 더 좋은 기본을 제공하기 위해서 변경될 때 발생하는 유지보수이다.

주요포인트 꼭 알고 가야해!

[외계인 코드(Alien Code)]

간단히 정의하면 15년 이전에 개발된 소스코드로 정의할 수 있다. 대부분 개발 시에 문서화를 하지 않았거나 개발자가 없거나 비구조적으로 작성한 경우들을 말한다. 이러한 외계인 코드는 유지보수를 상당히 어렵게 만드는 문제점 중에 하나이고 이러한 문제를 최소화하기 위해서는 유지보수를 고려한 표준화된 기술사용과 문서로 기록을 남겨야한다.

❖소프트웨어 형상 관리(Software Configuration Management)의 정의

- 소프트웨어에 대한 변경을 관리하기 위해 개발된 일련의 활동을 나타내며, 이런 변경이 전체 비용이 최소화되고 최소한의 방해가 소프트웨어의 현 사용자에게 야기되도록 보증하는 것을 목적으로 하는 것
- 개발 과정의 변화되는 사항을 관리
- 소프트웨어에서 일어나는 수정이나 변경을 알아내고 제어하는 것을 의미
- 소프트웨어에 가해지는 변경을 제어 관리하는 것

❖소프트웨어 형상관리 항목

- ① 정의 단계의 문서
- ② 개발 단계의 문서와 프로그램
- ③ 유지보수 단계의 변경 사항

❖소프트웨어 형상관리 목적

- ① 변경을 알아내기 위해
- ② 변경을 제어하기 위해
- ③ 변경이 적절히 수행되고 있는 것을 확인하기 위해
- ④ 변경 사항을 보고하기 위해

❖소프트웨어 유지보수와 SCM간의 차이점

- ① 유지보수는 소프트웨어가 고객에게 전달되고 운영중인 상태에서 발생하는 일련의 소프트웨어 공학활동이다.
- ② 소프트웨어 형상관리란 소프트웨어 프로젝트를 착수할 때 시작하는 추적과 제어 활동들의 집합이며, 소프트웨어의 작동이 잘못될 때 종료되는 활동이고, 개발 과정의 변화되는 사항을 관리한다.

주요포인트 꼭 알고 가야해!

주요포인트 꼭 알고 가야해!

(GO)

[07강] 객체지향, 프로젝트

1. 객체지향의 개념과 용어1
2. 객체지향의 개념과 용어2
3. 객체지향의 개발단계
4. 객체지향의 분석모델
5. 객체지향의 설계
6. 객체지향 프로그래밍과 테스트
7. 프로젝트 관리
8. 프로젝트 관리대상
9. 프로젝트 계획
10. 프로젝트 추정

합격
비법

대한민국 대표 IT강사
조 대 호

여러분의 인생드라마는
지금 잘 진행되시나요?
그 드라마의 주인공은
바로 당신입니다.
이 세상 모든사람들은
당신을 위한 조연입니다.
당신의 드라마가
성공한 그날!
당신의 감사메시지 속에서
저의 이름이 불려지기를
기원합니다.

정보처리기사/산업기사

합격은 당연! 고득점 합격이 목표!

(GO)

1. 객체지향의 개념과 용어1

JODAEHO

gisa plus

❖객체지향 개념

- 객체 지향 개념은 1980년대 초반 소프트웨어 위기를 맞게 되었을 때, 이를 극복하기 위해 등장하였다.
- 최근에 가장 널리 사용하는 기법으로, 사용자 중심의 대화식 프로그램 개발에 적합하다.
- 현실세계 영역을 프로그래머가 추구하는 자료 객체로 표현하며 세부적인 표현이 변경되어도 전체 시스템에 아무런 영향도 주지 않는 개발 기법이다.

❖객체(Object)

- 속성과 메서드를 캡슐화한 것으로 문제를 이해하는 기본 단위이다.

❖클래스(Class)

- 객체는 특정 시간, 특정 공간에 존재하는 실제적 엔티티이다. 클래스는 객체의 특성만 모아 놓은 추상(Abstraction)으로 그 근본이 된다. 객체의 유형 또는 타입(Object Type)이 클래스이다.

❖인스턴스(Instance)

- 클래스는 각각의 객체들의 타입이고, 객체는 메모리에 기억 장소가 할당된다. 각 클래스 타입에 따라 메모리 기억 장소가 할당된 객체를 그 클래스의 인스턴스라 부른다.

❖캡슐화(Encapsulation)

- 객체간의 통신을 위해 사용되는 인터페이스를 이용하여 각 모듈간의 독립성을 보장하기 위한 기법으로 객체를 이용하여 데이터와 연산들을 하나의 단위로 묶는 것을 말한다. 자료를 사용하기 위해서는 내부의 함수를 이용해서만 가능하다.
- 이점 : 재사용이 용이, 인터페이스가 단순, 변경 시 오류의 파급효과가 적다.

주요포인트 꼭 알고 가야해!

[속성(Attribute)]

- 객체 내의 데이터이며 객체의 현재상태를 나타낸다.

[메서드(Method)]

- 객체가 갖는 데이터를 처리하는 알고리즘
- 객체의 상태를 참조하거나 변경하는 수단.
- object로부터 메시지를 받을 때 시작
- 전통적 시스템의 function 또는 procedure에 해당하는 연산기능

정보처리기사/산업기사

합격은 당연! 고득점 합격이 목표!

(GO)

[07강] 객체지향, 프로젝트

Slides 48

❖정보 은닉(Information Hiding)

- 객체는 다른 객체들로부터 자신의 자료를 감추고, 자신의 연산을 통해서만 그 자료에 대한 접근을 허용하는 것을 말한다.

❖상속(Inheritance)

- 이미 정의되어 있는 상위 클래스의 메서드와 모든 속성을 하위 클래스가 물려받는 것을 말한다. 공유와 재사용성 향상이 목적이다.

❖다형성(Polymorphism)

- 하나의 메시지에 대해 각 클래스가 가지고 있는 고유한 방법으로 응답할 수 있는 능력을 말한다. 즉 한 메시지가 객체에 따라 다른 방법으로 응답 할 수 있는 능력을 말한다.

❖메시지 교환(Message Passing)

- 정의된 객체들 간의 정보교환단으로 다른 객체에 있는 함수 호출시 사용된다. 메시지를 받을 수신객체의 이름을 부르고, 작업 명령을 내리고 싶은 함수에 실행을 위한 매개 변수를 알려주고, 이를 실행시키는 것을 말한다.
- 메시지 형식 : <수신 객체> <함수 이름> <매개 변수>

❖추상 클래스(Abstract Class)

- 서브 클래스들의 공통된 특성을 추출하여 이것을 나타낸 클래스이다. 인스턴스를 만들 수 없다.
- 추상함수(Abstract Function 또는 Virtual Function) : 함수를 선언만 하고 구현은 서브 클래스로 미룬 함수를 말한다.
- 완전 클래스(Concrete Class) : 슈퍼 클래스의 추상 함수들을 모두 구현한 클래스를 말한다. 인스턴스를 만들 수 있다.
- 이점 : 재사용 부품을 이용하여 확장할 수 있게 하며 점진적인 개발에 도움을 준다.

주요포인트 꼭 알고 가야해!

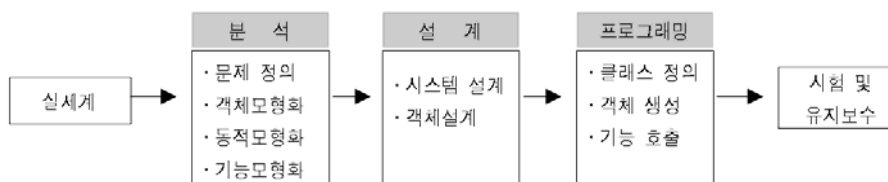
❖외부 인터페이스(Public Interface)

객체는 외부에서 접근 가능한 정보를 지정하고 이의 값에 접근하거나 변경하는 함수를 외부 인터페이스로 정의하고, 이를 통해서만 내부의 데이터에 접근할 수 있게 한다.

3. 객체지향의 개발단계

❖객체지향의 개발 단계

- 객체지향 분석(OOA : Object-Oriented Analysis)
 - 문제를 정의하고 이 정의로부터 모형들을 제작하여 실세계의 중요한 특성을 나타내는 단계
- 객체지향 설계(OOD : Object-Oriented Desing)
 - 분석단계의 객체 모형을 구현이 가능하도록 설계하는 단계
- 객체지향 프로그래밍(OOP : Object-Oriented Programming)
 - 설계 모형을 특정 프로그래밍 언어로 번역하는 단계



❖객체지향분석

- 문제의 정의로부터 시스템을 모델링하는 단계로 문제를 정의하고 이 정의로부터 객체 모델, 동적모델, 기능모델을 정의함으로써 실세계를 모델링한다.

❖객체지향 분석 작업을 하는 이유

- 재사용성을 높이기 위해서 한다.
- 공통된 속성을 명백히 표현한다.
- 한 객체와 다른 객체와의 종속성을 최소화할 수 있다.
- 소프트웨어 생명주기상에서 일관성을 유지할 수 있다.

주요포인트 꼭 알고 가야해!

❖객체 모델링(Object Modeling)

- 객체들을 식별하고 객체들 간의 관계를 정의하며 각 객체 클래스의 속성과 연산기능을 보여주는 시스템의 정적 구조(Static structure)를 나타낸다.
- 결과로 객체 다이어그램(Object Diagram)을 생성한다.
- 객체, 연산기능, 어트리뷰트를 식별한다.
- 객체들과 객체들 간의 관계를 정립한다.
- 상속관계를 정의하여 클래스들을 계층화 한다.
- 클래스들의 집단을 모듈로 정의한다.

❖동적 모델링(Dynamic Modeling)

- 상태 다이어그램을 이용하여 시스템의 동적인 행위를 표현하는 방법으로, 상태다이어그램에서 이용되는 주요 개념은 상태(State)와 사건(Event)이다.
- 사건은 하나의 객체로부터 다른 객체에 자극을 주는 것을 의미하며, 이 자극에 의해 객체의 상태가 변한다.
- 상태는 특정한 시점에서의 객체 속성 값을 의미하며, 이들은 시스템의 동적 구조인 동적 모델을 생성한다.
- 결과 : 상태전이도, 상태 추적 다이어그램

❖기능 모델링(Function Modeling)

- 자료 흐름도(DFD)를 이용하여 다수 프로세스들 간의 데이터 흐름을 중심으로 처리 과정을 표현한다.
- 자료 흐름도를 작성한다.
- 제어흐름과 제약조건을 파악한다.
- 각 절차 기능들을 정의한다.

주요포인트 꼭 알고 가야해!

[Coad와 Yourdon의 분석기법]

- E-R 다이어그램을 사용하여 엔티티(Entity)의 활동 등을 데이터 모델링하는데 초점을 둔 기법
- 구성요소 : 사람과 상호작용 요소, 문제영역요소, Task 관리요소, 데이터관리 요소

[럼바우(Rumbaugh)의 분석기법]

- 그래픽 표기법 이용
- 객체들의 연관을 강조, 조직적인 모델링 방법을 이용
- 분석절차 : 객체 모델링 → 동적 모델링 → 기능 모델링

5. 객체지향의 설계

❖객체 지향 설계의 개념

- 객체 지향 방법은 초기 단계인 분석, 설계 단계에서부터 클래스란 단위로 작업이 이루어지므로 소프트웨어의 개발 단계 사이에 의미적 차이가 거의 없다.
- 설계의 종류에는 시스템 설계와 객체 설계가 있다.

❖시스템 설계

- 시스템의 상위 구조를 선택하는 작업으로 전체적인 시스템 구조 즉, 어떤 서브시스템으로 구성되고, 구성된 서브시스템이 어떤 하드웨어에 할당될 것인가를 결정한다.
- 시스템 설계 단계에서 결정할 사항
 - 시스템을 서브시스템으로 분할하여 문제 안에 포함된 병행 수행 가능성을 파악한다.
 - 서브시스템을 프로세서와 태스크에 할당하고 자료 보관 방법을 결정한다.
 - 공동 자원의 접근 방법을 결정하고 제어에 대한 구현 방법을 결정한다.
 - 경계 조건 처리 방법을 정한후 우선순위를 정한다.

❖객체 설계

- 구현에서 사용될 클래스와 클래스 안의 오퍼레이션, 클래스 사이의 관계를 구현하기 위한 알고리즘을 완벽하게 정의한다.
- 분석 단계에서 작성한 모형을 더 구체화하되 이전까지의 응용 분야의 관점에서 컴퓨터 구현으로 관점을 전환한다.
- 객체 설계 순서
 - 세 가지 모형을 이용하여 객체에 대한 오퍼레이션을 파악한다.
 - 오퍼레이션을 구현하기 위한 알고리즘을 정하고 자료에 대한 접근 경로를 최적화한다.
 - 외부와 상호작용하기 위한 제어를 구현한 후 상속성을 높이기 위하여 클래스 구조를 조정한다.
 - 관계를 설계하고, 객체의 표현 방법을 결정한다.
 - 클래스와 관계를 단일 모듈로 패키지화한다.

주요포인트 꼭 알고 가야해!

[Booch의 객체지향 설계]

- DFD를 사용
- 객체를 분해, 객체들간의 인터페이스를 찾아 이것은 Ada 프로그램으로 변환

❖객체 지향 프로그래밍

- 클래스를 정의하고 객체를 생성한다.
- 기능을 호출하다음 상속개념을 활용한다.
- 객체 클래스간에 상호 작용하게 한다.

❖객체 지향 테스트

- 객체 지향 시스템을 적합하게 테스트하려면 단위 테스트(Unit Testing) → 통합 테스트(Integration Testing) → 검증과 시스템 테스트(Validation And System Testing)으로 수행해야 한다.
- 테스팅의 정의에 객체지향 환경에 적용 가능한 오류 발견 기법이 포함되도록 확장되어야 한다.

❖단위 테스트(Unit Testing)

- 테스트할 수 있는 가장 작은 단위는 캡슐화 또는 클래스 또는 객체이다.
- 하나의 연산만을 따로 분리해서 테스트하는 것보다는 클래스의 일부로서 테스트해야 한다.

❖통합 테스트(Integration Testing)

- 쓰레드-기저 테스트(Thread-Based Testing) : 시스템에 대한 하나의 입력 또는 이벤트에 응답하는데 요구되는 클래스의 집합을 통합한다.
- 사용-기저 테스트(Use-Based Testing) : 통합 접근법으로 서버 클래스에 독립되어 있는 클래스(Independent Class)를 테스트해서 시스템 구축을 시작한다.

❖검증 테스트(Validation Testing)

- 시스템로부터 사용자가 인식할 수 있는 출력과 사용자의 가시적 행동에 초점을 맞춘다.
- 분석 모형의 일부분인 Use Case를 사용하여 테스트한다.
- 검증 테스트를 유도하는데 블랙-박스 테스트 방법을 사용할 수 있다.

주요포인트 꼭 알고 가야해!

7. 소프트웨어 프로젝트관리

❖소프트웨어 프로젝트 관리란?

- 프로젝트 관리는 단계나 활동이라기보다는 소프트웨어 공학 과정의 첫 번째 계층이다.
- 프로젝트 관리는 소프트웨어 생명 주기의 전 과정에 걸쳐 진행된다.
- 효과적 프로젝트 관리를 수행하기 위해서는 수행될 작업의 영역, 당하게 될 위험성, 요구되는 자원, 성취해야 할 작업, 추적해야 될 이정표, 확대될 노력, 따라야 될 일정 등을 이해해야 한다.

❖사람(People)

- 담당자 : 프로젝트 관리에 있어서 가장 중요하며 프로젝트 담당자, 팀장, 소프트웨어 팀 등을 고려.
 - 선임 매니저 : 프로젝트의 방향을 설정하며, 프로젝트 매니저는 프로젝트의 계획, 작성, 동기부여, 조직 구성, 실무자를 관리한다.
 - 프로젝트 매니저 : 개발 계획을 만들고, 개발 동기를 부여하고, 개발 팀을 구성하고, 실제 소프트웨어를 만들 실무자들을 관리하는 사람.
 - 실무자 : 소프트웨어제품과 Application을 공학화 시키는데 필요한 실무 기술을 담당.
 - 고객 : 소프트웨어에 대한 요구사항을 제시해주는 사람.
 - 최종 사용자 : 고객이 요구한 소프트웨어를 직접 사용한다.
- 팀장
 - 팀장이 추구해야 할 점 - MOI 모델 : 동기부여(Motivation), 조직화(Organization), 아이디어나 기술 혁신(Idea or Technical innovation)
 - 능력 있는 관리자가 갖추어야 할 특성 : 문제 해결 능력, 관리 식별 능력, 성취 의욕, 팀 구축능력
- 소프트웨어 팀
 - 프로젝트에 몇 명의 사람을 투입하고 그 조직을 어떻게 구성하는 것이 최상의 효과를 만들 것인가는 관리자의 능력이다. '최상의' 팀 구조는 조직의 관리 방식과 팀을 구성하는 인원수, 그들이 소유하고 있는 기술 수준 및 해결해야 하는 문제의 난이도에 따라 결정된다.

주요포인트 꼭 알고 가야해!

프로젝트관리 대상(3P)

People (사람) : 인적자원
 Problem (문제) : 문제인식
 Process (처리) : 작업계획

❖조직의 구성방식(Mantei 제시)

팀 유형	DD (민주적 분산화)	CD (통제적 분산화)	CC (통제적 집중화)
작업 난이도	높을 때 적합	낮을 때 적합	낮을 때 적합
프로젝트 규모	소규모에 적합	대규모에 적합	대규모에 적합
팀 존속 기간	장기간에 적합	단기간에 적합	단기간에 적합
신뢰성(개발능력)	높을 때 적합	높을 때 적합	낮을 때 적합
사교성(의사교환)	높을 때 적합	낮을 때 적합	낮을 때 적합
모듈화	낮을 때 적합	높을 때 적합	높을 때 적합

❖문제(Problem)

- 프로젝트를 계획하기 전에 프로세스의 목적과 영역이 설정되어야 한다. 발생할 수 있는 문제점을 고려하고, 기술과 관리상의 제약 사항들을 정확히 식별해야 한다. 이러한 작업들은 비용의 정확한 측정, 위험적인 요소의 효과적인 대처 방안, 진행의 효과적인 지표를 정할 수 있게 된다.

❖프로세스

- 소프트웨어 프로세스는 소프트웨어 개발에 필요한 골격을 제공한다. 프로세스를 개선하는 방법은 프로세스의 특정 속성들을 측정하고, 이를 근거로 프로젝트를 개선한다. 프로세스는 소프트웨어 품질과 조직의 성능에 크게 영향을 미치는 3개의 요소, 즉 제품, 사람, 기술 등을 고려해야 한다.

주요포인트 꼭 알고 가야해!

9. 프로젝트 계획

❖계획 수립 목적

- 관리자가 개발 과정에 소요되는 비용, 자원 그리고 일정 계획의 모든 정보를 모아서 합리적인 예측을 하기 위한 기초를 만드는데 있다. 그럼으로 개발과정에서 발생할 수 있는 여러 가지 위험요인을 최소화 할 수 있다.

❖소프트웨어 영역(Software Scope)

- 프로젝트 계획 시 첫 번째 해야 할 일은 소프트웨어 영역을 선택하는 일이다. 이 영역 속에는 기능(Function), 성능(Performance), 제약(Constraint), 개발인원, 일정계획, 인터페이스 그리고 신뢰성 등이 포함된다. 기능 영역에는 비용과 스케줄, 분해의 정도가 포함되고, 성능 영역에는 프로세싱과 응답 시간 등이 포함되고, 제약 영역에는 외부 하드웨어, 사용 가능한 메모리 등이 포함될 수 있다.

❖자 원

- 계획 수립의 두 번째 단계는 개발에 사용될 자원을 예측하는 것이다. 이 자원에는 개발 환경, 재사용이 가능한 소프트웨어 컴포넌트 그리고 사람으로 구성된다.
- 인적 자원 : 계획의 책임자는 조직의 직위를 구성해야하며, 작은 프로젝트일 경우에는 한 개인만으로도 조직을 만들 수도 있다.
- 재사용 가능 소프트웨어 자원
 - Off-the-Shelf Components : 개발 업체에서 얻을 수 있거나 내부적으로 개발했던 소프트웨어, 현재의 작업에 사용될 수 있으며, 이미 검증이 된 요소
 - Full-Experience Components : 과거에 현재와 유사한 프로젝트를 개발하면서 만들어진 코드, 설계서 등. 현재의 개발 인원은 이 요소들에 대해서 잘 알고 있고, 전체적으로 이 요소들에 대한 수정에 많은 노력이 필요 없다.
 - Partial-Experience Components : 과거에 현재와 유사한 프로젝트를 개발하면서 만들어진 설계서 등을 말한다. 그러나 현재의 개발자들이 이 요소에 대해서 부분적으로 알고 있으며, 자원의 수정이 많으며 또한 위험성이 크다.
 - New Components : 현재의 프로젝트에서 새로이 만들어야 하는 요소들
- 주변 자원
 - 소프트웨어 개발 시 프로젝트를 도와주는 하드웨어 및 소프트웨어

주요포인트 꼭 알고 가야해!



주요포인트 꼭 알고 가야해!

(GO)

[08강] S/W공학의 발전주제

1. 프로젝트 추정
2. 자동화 도구
3. 프로젝트 일정
4. 프로젝트 일정계획
5. 소프트웨어 품질보증
6. 소프트웨어 검토
7. 위험관리와 재사용
8. 재공학과 SERVER/CLIENT
9. CASE

합격
비법대한민국 대표 IT강사
조 대 호

여러분의 인생드라마는
지금 잘 진행되시나요?
그 드라마의 주인공은
바로 당신입니다.
이 세상 모든사람들은
당신을 위한 조연입니다.
당신의 드라마가
성공한 그날!
당신의 감사메시지 속에서
저의 이름이 불려지기를
기원합니다.

정보처리기사/산업기사

합격은 당연! 고득점 합격이 목표!

(GO)

1. 프로젝트 추정

JODAEHO

gisa plus

❖추정이란?

- 소프트웨어를 개발하기 전에 개발 될 소프트웨어에 대해서 발생할 여러 상황을 미리 생각하는 것이며 추정은 말대로 미래에 대한 예측이므로 항상 불확실한 위험이 내재하고 있다.

❖소프트웨어 측정

- 직접 측정 - 생산된 LOC, 실행속도, 메모리 크기, 어떤 시간 동안에 발견된 결함 등
- 간접 측정 - 기능성, 품질, 복잡도, 효율성, 신뢰성, 유지보수성 등
- 크기-중심 매트릭스
 - 소프트웨어와 소프트웨어가 개발되는 과정을 직접적으로 측정하는 방법으로 크기를 고려해서 품질 및 생산성 측정을 정규화 시켜 유도된다.
- 기능-중심 매트릭스 : FP(Function Point) 사용
 - 소프트웨어와 소프트웨어가 개발되는 과정을 간접적으로 측정하는 방법으로 정규화 된 값으로 애플리케이션이 전달하는 기능성 측정에 사용된다.

❖소프트웨어 추정 모형 - COCOMO(Constructive COst Model)

- 소프트웨어의 종류에 따라 다르게 책정되는 비용 산정방정식을 이용하는 합리적인 방법으로 Barry Boehm이 제시한 원시 프로그램의 규모에 의한 비용 경험적 추정모형.

❖Basic COCOMO모형의 모드(Mode)

- 기본형(Organic mode) : 기관 내부에서 개발된 중소 규모의 소프트웨어로 일괄 자료 처리나 과학 기술 계산용, 비즈니스 자료 처리용으로 5만 라인 이하의 소프트웨어를 개발하는 유형이다.
- 중간형(Semidetached mode) : 요구 사항이 중간 정도로 종합적인 경험을 갖고 있는 팀이 수행하는 중.소규모의 트랜잭션 처리 시스템이나 운영체제, 데이터베이스 관리시스템 등의 30만 라인 이하의 소프트웨어를 평가하는 유형이다.
- 내장형(Embedded mode) : 최대형 규모의 트랜잭션 처리 시스템이나 운영체제 등의 소프트웨어를 개발하는 유형이다.

주요포인트 꼭 알고 가야해!

정보처리기사/산업기사

합격은 당연! 고득점 합격이 목표!

(GO)

[08강] S/W공학의 발전적주제

Slides 58

❖COCOMO 모형의 계층

- 기본형(Basic) COCOMO : 프로그램 크기의 함수가 추정된 LOC로 표현해서 소프트웨어 개발노력을 계산하는 고정 단일 값 모형
- 중간형(Intermediate) COCOMO : 프로그램 크기의 함수와 제품, 하드웨어, 인적요소, 프로젝트 속성들의 주관적인 요소의 평가를 내포하고 있는 비용 유도자의 집합으로 계산
- 고급형(Detailed) COCOMO : 소프트웨어 공학 과정의 각 단계에 비용 유도자의 영향을 평가하는 중간 형식의 모든 특성을 통합 시킨 것

❖Putnam의 생명주기 예측모형

- 소프트웨어개발 프로젝트의 생명주기 전 과정동안에 노력의 특수한 분포를 가정해 주는 동적 다변수 모형

❖Function Point(기능점수) 모형

- 소프트웨어의 생산성을 측정하기 위해 소프트웨어기능을 증대시키는 요인별로 가중치를 부여하고 그 값을 환산하여 기능점수를 구하는 것
- 기능점수는 소프트웨어개발의 규모, 복잡도, 난이도 등을 숫자로 나타낸 것

❖자동화 추정 도구

- COCOMO 기초 : BYL, WICOMO, DECPlan
- Putnam 기초 : SLIM
- FP 추정방법 : ESTIMACS

❖소프트웨어 프로젝트의 예측방법

- 프로젝트에서 예측을 가능한 뒤로 미룬다.
- 프로젝트 비용과 노력측정을 위해 상대적으로 간단한 분해기술을 이용.
- 소프트웨어 비용과 노력에 대한 경험적 측정 모형을 활용한다.

주요포인트 꼭 알고 가야해!

COCOMO의 비용 산정 예
개발에 소요되는 노력이 40PM(Programmer-Month)으로 계산되었다. 개발에 소요되는 기간이 5개월이고, 1인당 인건비가 100만원이라면 이 프로젝트에 소요되는 총 인건비는 얼마인가?

3. 프로젝트 일정

❖프로젝트 일정이란?

- 프로젝트 일정이란 현재 진행 중인 프로젝트에 들어갈 노력을 프로젝트 기간에 고르게 분배하는 활동이다. 프로젝트 초기에는 포괄적인 스케줄이 적용되며, 이때에는 만들 제품에 대한 기능적인 면을 검토하게 되고, 프로젝트 진행 중에는 각각의 항목에 대한 세부 스케줄로 변하게 된다.

❖기본 원칙

- 구획화(Compartmentalization) : 프로젝트는 가능한 많은 부분으로 나누어져야 한다.
- 상호의존(Interdependency) : 나누어진 그룹끼리의 역할 및 상관관계를 정해야 한다.
- 시간 할당(Time Allocation) : 나누어진 그룹에 대한 시작 및 종료시점, Full-Time으로 할지, Part-Time으로 할지를 정해야 한다.
- 노력 검증(Effort Validation) : 프로젝트에 투입되는 인원은 정해져 있고, 한정된 인원이 정해진 시간에 알맞게 프로젝트에 참여했는지를 확인해야 한다.
- 책임성(Defined Responsibility)
- 정의된 결과물(Defined Outcome) : 프로젝트에 대한 결과물을 확인해야 한다.
- 계획된 이정표(Defined Milestone)

❖Brooks의 법칙(사람-작업 관계)

- 만약 기획된 일정에 뒤쳐져 있더라도 더 많은 프로그래머를 투입시키면 후반부에서는 계획된 일정을 따라 잡을 수 있다. → 잘못된 개념
- 후반부에 참여하는 인원들은 오히려 일정을 더 지연시키는 원인이 된다.

❖노력 분포도(40-20-40 규칙)

- 개발에 드는 노력은 분석 및 설계에 40%, 코딩은 20%, 나머지 40%에는 디버깅이 차지한다.

주요포인트 꼭 알고 가야해!

❖일정 계획(스케줄링)의 방법

- WBS(Work Breakdown Structure) : 작업 분할 구조
 - 모든 개발 단계를 서로 연관된 소 작업으로 분류하고 이들의 시작부터 끝나는 시점까지의 관계를 분석하여 개발기간 및 개발비용을 측정하는 방법이다. 모든 작업을 트리 구조로 분할 한 후 단노드부터 일정을 측정하면서 전체 일정을 잡는다.
- 간트 차트(Gantt Chart)
 - 프로젝트의 세부 사항들을 병렬로 보이도록 일정에 따라 막대그래프로 표시한다.
 - 막대의 길이는 시간을 의미하며, 각 작업의 시작 시점과 종료 시점을 보여준다.
 - 간트 차트 내용 : 생명주기 단계, 일정계획(작업일정), 이정표, 작업기간
- PERT/CPM : 프로그램 평가와 검토 기법
 - 프로젝트의 지연을 방지하고 계획대로 진행되게 하기 위한 일정계획의 방법으로 대단위 계획의 조직적인 추진을 위해 자원의 제약하에 비용을 적게 사용하면서 초단시간내 계획 완성을 위한 프로젝트 일정 방법
 - 프로젝트 작업 사이의 관계를 나타내며 최장경로를 파악할 수 있다.
 - 가장 큰 목적은 전체 프로젝트 개발기간을 결정하는 임계경로를 파악하는데 있다.
- PERT/CPM이 제공하는 도구
 - 프로젝트 개발기간을 결정하는 임계경로
 - 통계적 모델을 적용해서 개별 작업의 가장 근접한 시간 측정 기준
 - 정의작업에 대한 시작시간을 정의하여 작업들 간의 경계시간 계산

❖일정계획에 필요한 작업

- ① 프로젝트의 규모를 추정한다.
- ② 각 단계에 필요한 작업들을 분리한다.
- ③ 각 작업의 상호의존 관계를 CPM 네트워크로 나타낸다.
- ④ 일정계획을 간트 차트로 나타낸다.

주요포인트 꼭 알고 가야해!

5. 소프트웨어 품질보증

❖소프트웨어 품질 보증(SQA : Software Quality Assurance)

- 소프트웨어 품질보증이란 어떤 항목이나 제품의 설정된 기술적인 요구사항과 일치하는가를 적절하게 확인하는데 필요한 체계적이고도 계획적인 유형의 활동이다.
- 소프트웨어 품질보증의 일반적인 개념은 사용자가 요구하는 품질이 충분히 충족되고 있는 것을 보증하기 위한 생산자가 구성하는 체계적인 활동으로 정의되고, 품질의 정의 및 측정, 품질의 정도, 판단 등이 주요 관심사가 된다.

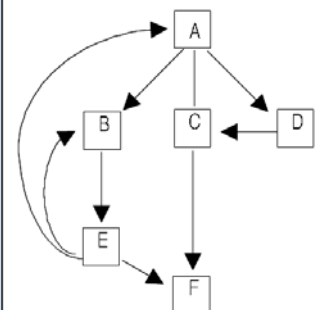
❖소프트웨어 품질 척도 방법

- Halstead의 소프트웨어 과학 매트릭스
- Macabe의 소프트웨어 복잡도 매트릭스
 - 사이클로매틱(Cyclomatic : 순환적)복잡도에 기초하여 정의
 - 그래프를 이용하여 프로그램의 제어흐름을 표시
 - 복잡도(V) = 화살표수(A) - 노드수(N) + 2
 - ⇒ 복잡도는 그래프의 영역의 수와 일치한다.
 - 영역의 수 = 폐쇄된 지역 + 그래프의 바깥쪽(1)

복 잡 도	평 가
5 미만	매우 간단한 P/G
5 - 10	매우 구조적이며 안정된 P/G
20 이상	문제 자체가 복잡하거나 구조가 필요이상으로 불안정한 P/G
50 이상	매우 비구조적이며 불안정한 P/G

- 라인수(LOC) 매트릭스
 - 인월당 산출되는 P/G 라인수를 소프트웨어 생산의 척도로 삼는다.
- FP 매트릭스
 - FP를 소프트웨어 생산의 척도로 가짐, 가장 유용한 모형으로 평가
 - 데이터의 복잡도와 P/G간의 구조적 연관성을 복합적으로 고려

주요포인트 꼭 알고 가야해!



- 모듈의 최대 크기의 수량적인 지표를 제공
- V(G) = 모듈 크기의 실질적인 상한선

❖소프트웨어 검토란?

- 소프트웨어 검토는 소프트웨어 개발 시 여러 단계에서 실시되며, 각 단계마다 제거 할 수 있는 오류들을 발견하는데 사용한다. 즉, 개발 과정에서 필요한 부분과 필요 없는 부분을 구분하는 역할을 한다.

❖결함 증폭 모형(Defect amplification model)

- 소프트웨어 공학과정의 예비설계, 상세 설계, 그리고 코딩단계 동안에 오류의 발생과 탐지를 설명하는데 사용되며 전 단계에서 나온 오류는 다음 단계의 작업에서 증폭된다.

❖정형 기술 검토(FTR : Formal Technical Review)

- 소프트웨어 엔지니어들이 수행하는 대표적인 품질보증 활동으로 소프트웨어 결함을 조기에 발견하여 다음 개발단계로 넘어가지 전에 결함을 수정하도록 한다. 구조적 검토 회(Walkthroughs), 검열(Inspection), 라운드로빈 검토 등을 사용하는 소프트웨어 검토의 한 부분이라 할 수 있다.
- 목적
 - 프로젝트를 보다 관리하기 쉽게 만들며 일관된 방법으로 개발 되도록 한다.
 - 미리 정한 기준에 따라 표현되도록 하고 최종 결과물의 요구 사항과 일치하는지를 검증하며 표현에 대한 기능, 논리 또는 구현에서 오류를 발견한다.
- 검토 후 결정해야 할 항목들
 - 더 이상 변경 없이 제품을 사용할 것인가?
 - 심각한 오류로 인해 제품 개발을 거부할 것인가?
 - 약간의 오류는 있지만 임시적으로 제품을 수용할 것인가?
- 검토 요약 보고서
 - 검토 회의가 끝나고 회의 내용을 요약해서 검토 요약 보고서를 작성한다. 이 보고서에는 "(1) 무엇을 검토했는가, (2) 누가 검토했는가, (3) 검토 결과와 결론은 무엇인가"에 대한 대답이 포함되어 있어야 한다.

주요포인트 꼭 알고 가야해!

7. 위험관리와 재사용

❖위험 관리란?

- 프로젝트에 내재된 위험 요소를 파악하고 그 영향을 조사하여 관리하는 활동이 위험 관리이다. 즉, 프로젝트를 성공적으로 수행하기 위한 위험 요소를 미리 예측하여 준비하는 모든 기술과 활동을 말한다.

❖위험의 범위

- 프로젝트 위험(Project Risk) : 이 위험이 발생되면 계획은 지연되고 추가 비용은 증가한다. 프로젝트의 복잡도, 크기 그리고 구조 불확실 정도에 따라서 영향을 받는다.
- 기술 위험(Technical Risk) : 이 위험은 프로젝트를 구현 시 기술적으로 한계를 느꼈을 때 발생하며, 이 위험이 증가되면 프로젝트 자체를 실패로 이끌 수도 있다. 이것의 구성 요소로는 명세의 모호성, 기술의 불확실성, 기술의 쇠퇴 그리고 첨단 기술 등으로 이루어진다.
- 비즈니스 위험(Business Risk) : 프로젝트의 결과물을 팔려고 할 때 발생하는 것으로 소프트웨어의 생존 능력을 위협한다.

❖위험관리의 일반적인 절차

- 위험식별 → 위험분석 및 평가 → 위험관리계획 → 위험감시 및 조치

❖소프트웨어 재사용이란?

- 기존의 기능 및 품질을 인정받은 소프트웨어의 전체 혹은 일부분을 재사용하여 새로 개발되는 소프트웨어의 질을 높이고 생산성을 향상시켜 개발시간과 비용을 감소시키는 소프트웨어 위기의 해결책이다.

❖재사용의 효과

- 개발 시간과 비용 감소, 소프트웨어의 품질 향상, 생산성의 증가
- 프로젝트의 실패 위험을 줄여주며 재사용 부품의 크기가 작을수록 재사용 확률이 높다.
- 시스템과 시스템 개발에 관한 지식의 공유

주요포인트 꼭 알고 가야해!

❖재공학이란?

- 소프트웨어 위기의 해결책을 개발의 생산성이 아닌 유지보수 문제에서 찾으려는 새로운 시각을 말한다. 기존 소프트웨어의 취약한 부분들을 단계적으로 미화시켜 작업 수행 시마다 질적 향상을 꾀하는 작업을 말한다.

❖재공학의 활동

- 분석(Analysis) : 소프트웨어 성분들과 프로그램 동작을 이해하고, 재공학 대상을 선정하거나 시스템 품질 평가를 위하여 기존 시스템의 원시코드와 문서화를 조사·분석하는 활동.
- 재구성(Restructuring) : 소프트웨어 기능은 수정하지 않고 개발 프로젝트의 정의, 데이터, 원시 코드 등의 형태만 바꾸는 활동.
- 역공학(Reverse Engineering) : 원시코드를 분석하여 소프트웨어의 성분들과 그들 간의 관계를 생성해내는 과정으로 가장 하위 수준의 코드로부터 상위 수준의 분석·설계 정보를 다시 만들어내는 활동이다.
- 이주(Migration) : 기존 소프트웨어 시스템을 다른 프로그래밍 언어나 운영체제, 새로운 기술 또는 하드웨어 환경으로 변환하는 활동.

❖Client/Server 소프트웨어 공학

- 클라이언트/서버 시스템의 구조
 - 클라이언트/서버 시스템은 네트워크, 데이터 통신, 분산표현, 분산처리, 표준안, 개방 시스템 등의 여러 가지 요소를 잘 조화시켜 모델을 구성하는 분산처리 구조의 한 형태이다.
- 클라이언트(Client) : 클라이언트는 시스템(서버)에 서비스를 요청한다.
- 서버(Server) : 서버는 시스템에서 서비스를 제공한다.

주요포인트 꼭 알고 가야해!

[3R]

Re-use
Re-engineering
Reverse-Engineering

9. CASE

❖CASE(Computer-Aided Software Engineering)의 정의 및 장점

- 소프트웨어 개발의 자동화 도구.
- 소프트웨어 개발 과정에서 사용되는 요건 분석, 설계, 구현, 검사 및 디버깅 과정에서 컴퓨터와 전용의 소프트웨어 도구를 사용하여 자동화하는 작업
- 전체 소프트웨어의 생산성과 신뢰성 문제를 해결할 수 있다.
- 정형화된 구조 및 메커니즘을 소프트웨어 개발에 적용하여 소프트웨어 생산성 향상을 구현하는 공학기법이다.
- 소프트웨어의 개발과정을 자동화함으로써 생산성을 증대시키고자 하는 목적으로 개발되었다.
- CASE를 사용함으로써 개발의 표준화를 지향하고, 자동화의 이점을 얻을 수 있다.
- 개발 기간이 짧고 개발비용이 절감된다.
- 프로그램의 유지 보수를 간편하게 한다.

❖CASE의 분류

- 상위(Upper) CASE
 - 문제를 기술하고 계획하며 요구 분석을 수행하는 도구로 명세서를 작성하고 설계하는 과정에 쓰인다.
 - 종 류 : 자료 흐름도, E-R 다이어그램, 상태 전이도, 자료 사전
- 하위(Lower) CASE
 - 코드의 작성과 테스트, 문서화하는 과정을 수행하는 도구들을 포함한다.
 - 종 류 : 코드 생성기, 구문 중심 편집기, 전문가 시스템을 이용한 컴파일러, 테스트 도구, 컴파일러, 링커, 디버거 등
- 통합(Integrated) CASE
 - 소프트웨어 개발 주기에 포함되는 전체 과정을 지원하기 위한 도구이다. 공통의 정보 저장소와 통일된 사용자 인터페이스를 사용하여 도구들을 통합한다.
 - 종 류 : 그래픽 기능, 프로토타이핑 및 명세화 도구, 설계 도구, 프로그래밍 및 테스트 도구, 정보 저장소 등

주요포인트 꼭 알고 가야해!