

Char10. 응용 SW 기초 기술 활용

운영체제 OS Operating System



- 컴퓨터 시스템의 자원들을 효율적으로 관리하며, 사용자가 컴퓨터를 편리하고 효율적으로 사용할 수 있도록 환경을 제공하는 여러 프로그램의 모임
- 컴퓨터 사용자와 컴퓨터 하드웨어 간의 인터페이스로서 동작하는 시스템 소프트웨어의 일종

운영체제의 목적

처리 능력 Throughput 향상	일정 시간 내에 시스템이 처리하는 일의 양
반환 시간 Turn Around Time 단축	시스템에 작업을 의뢰한 시간부터 처리가 완료될 때까지 걸린 시간
사용가능도 Availability 향상	시스템을 사용할 필요가 있을 때 즉시 사용 가능한 정도
신뢰도 Reliability 향상	시스템이 주어진 문제를 정확하게 해결하는 정도

운영체제의 기능

- 프로세서, 기억장치, 입출력장치, 파일 및 정보 등의 자원을 관리
- 자원을 효율적으로 관리하기 위해 자원의 스케줄링 기능을 제공
- 사용자와 시스템 간의 편리한 인터페이스를 제공
- 시스템의 각종 하드웨어와 네트워크를 관리 및 제어
- 데이터를 관리하고, 데이터 및 자원의 공유 기능을 제공
- 시스템의 오류를 검사하고 복구한다.
- 자원보호 기능을 제공
- 입출력에 대한 보조 기능을 제공
- 가상 계산기 기능을 제공

운영체제의 주요 자원 관리

프로세스 관리	프로세스 스케줄링 및 동기화 관리 담당 프로세스 생성과 제거, 시작과 정지, 메시지 전달 등의 기능을 담당
기억장치 관리	프로세스에 메모리 할당 및 회수 관리 담당
주변장치 관리	입출력장치 스케줄링 및 전반적인 관리 담당
파일 관리	파일의 생성과 삭제, 변경, 유지 등의 관리 담당

운영체제 종류

MS-DOS	단일 작업 처리 시스템	개인용 운영체제
Windows	다중 작업 처리 시스템	
MacOS		
UNIX		서버용 운영체제
LINUX		

Windows

1. 그래픽 사용자 인터페이스 GUI Graphic User Interface

키보드로 명령어를 직접 입력하지 않고 마우스로 아이콘이나 메뉴를 선택하여 모든 작업을 수행하는 방식

2. 선점형 멀티태스킹 Preemptive Multi-Tasking

동시에 여러 개의 프로그램을 실행하는 멀티태스킹을 하면서 운영체제가 각 작업의 CPU 이용시간을 제어하여 응용 프로그램 실행 중 문제가 발생하면 해당 프로그램을 강제 종료 시키고 모든 시스템 자원을 반환하는 방식

3. PnP Plug and Play 자동 감지 기능

컴퓨터 시스템에 프린터나 사운드 카드 등의 하드웨어를 설치했을 때 해당 하드웨어를 사용하는데 필요한 시스템 환경을 운영체제가 자동으로 구성해 주는 기능

4. OLE Object Linking and Embedding

다른 여러 응용 프로그램에서 작성된 문자나 그림 등의 개체를 현재 작성 중인 문서에 자유롭게 연결하거나 삽입하여 편집할 수 있게 하는 기능

5. 255자의 긴 파일명

Windows에서 파일 이름은 ₩ / * ? " < > | 를 제외한 모든 문자 및 공백을 이용하여 최대 255자 까지 지정 가능

6. Single-User 시스템

컴퓨터 한 대를 한 사람만이 독점하여 사용

UNIX

1. 시분할 시스템을 위해 설계된 대화식 운영체제

커널 Kernel

2. 소스가 공개된 개방형 시스템

UNIX의 가장 핵심적인 부분

3. 대부분 C언어로 작성되어 있어 이식성이 높으며 장치, 프로세스 간의 호환성이 높다.

프로그램과 하드웨어 간의 인터페이스 역할

4. 크기가 작고 이해하기 쉽다.

프로세스 관리, 기억장치 관리, 파일 관리, 입출력 관리, 프로세스간 통신, 데이터 전송 및 변환 등 여러 기능 수행

5. 다중 사용자 Multi-User, 다중 작업 Multi-Tasking을 지원

셸 Shell

사용자의 명령어를 인식하여 프로그램을 호출하고 명령을 수행하는 명령어 해석기

6. 많은 네트워킹 기능을 제공하므로 통신망 관리용 운영체제로 적합

시스템과 사용자 간의 인터페이스를 담당

7. 트리 구조 Tree의 파일 시스템을 갖는다.

DOS의 COMMAND.COM과 같은 기능 담당

8. 전문적인 프로그램 개발에 용이하다.

종류 : Bourne Shell, C Shell, Korn Shell 등

9. 다양한 유틸리티 프로그램들이 존재

유틸리티 프로그램 Utility Program

일반 사용자가 작성한 응용프로그램을 처리하는 데 사용

DOS에서의 외부 명령어에 해당

운영체제 제어하는 방법

1. CLI Command Line Interface

2. GUI

Windows 명령어	명령 프롬프트 CMD 창에 명령어 입력
DIR	dir 현재 디렉터리의 파일 목록 표시
COPY	copy abc.txt gilbut abc.txt 파일을 gilbut 디렉터리에 복사
DEL	del abc.txt abc.txt 파일을 삭제
TYPE	type abc.txt abc.txt 파일의 내용을 표시
REN	ren abc.txt 123.txt abc.txt 파일의 이름을 123.txt로 변경
MD	md gilbut gilbut 디렉터를 생성
CD	cd gilbut 디렉터리 위치를 gilbut으로 변경
CLS	cls 화면에 표시되어 있는 모든 내용을 삭제
ATTRIB	attrib +r abc.txt abc.txt 파일의 속성을 읽기 전용으로 변경
FIND	find "123" abc.txt abc.txt에서 "123"이 포함된 문자열을 찾는다.
CHKDSK	chkdsk 현재 드라이브의 상태를 점검
FORMAT	format C: C 드라이브를 초기화
MOVE	move abc.txt gilbut abc.txt 파일을 gilbut디렉터리로 이동

UNIX/LINUX 명령어	셸 Shell에 명령어 입력
cat	cat abc.txt abc.txt파일 내용을 화면에 출력
cd	cd gilbut gilbut 디렉터리로 이동
chmod	chmod u=rwx abc.txt User(s)에게 abc.txt파일의 읽기, 쓰기, 실행 권한을 부여(=)
chown	chown member1 abc.txt abc.txt파일의 소유자를 member1로 변경
cp	cp abc.txt gilbut/abc2.txt abc.txt파일을 gilbut디렉터리에 abc2.txt로 이름을 변경하여 복사
rm	rm abc.txt abc.txt파일 삭제
find	find abc.txt abc.txt파일 찾기
fsck	fsck /dev/sda1 /dev/sda1에 기록된 모든 파일 시스템을 검사하고 보수
kill	kill1234 PID가 1234인 프로세스를 종료
killall	killall gilbut 프로세스 이름이 gilbut인 모든 프로세스 종료
ls	ls 현재 디렉터리의 파일 목록 표시
mkdir	mkdir gilbut gilbut 디렉터리 생성
rmdir	rmdir gilbut gilbut 디렉터리 삭제
mv	mv abc.txt gilbut/abc2.txt abc.txt 파일을 gilbut 디렉터리에 abc2.txt로 이름을 변경하여 이동
ps	ps 현재 실행중인 프로세스 표시
pwd	pwd 현재 작업중인 디렉터리 경로를 화면에 표시
top	top 시스템의 프로세스와 메모리 사용 현황을 표시
who	who 현재 시스템에 접속해 있는 사용자를 표시

기억장치 관리 전략 종류

1. 반입 전략 Fetch

보조기억장치에 보관중인 프로그램이나 데이터를 언제 주기억장치로 적재할 것인지 결정하는 전략

- 요구 반입 Demand Fetch

실행중인 프로그램이 특정 프로그램이나 데이터 등의 참조를 요구할 때 적재하는 방법

- 예상 반입 Anticipatory Fetch

실행중인 프로그램에 의해 참조될 프로그램이나 데이터를 미리 예상하여 적재하는 방법

2. 배치 전략 Placement

새로 반입되는 프로그램이나 데이터를 주기억장치의 어디에 위치시킬 것인지 결정하는 전략

- 종류: 최초 적합 First Fit, 최적 적합 Best Fit, 최악 적합 Worst Fit

3. 교체 전략 Replacement

주기억장치의 모든 영역이 이미 사용중인 상태에서 새로운 프로그램이나 데이터를 주기억장치에 배치하려고 할 때, 이미 사용되고 있는 영역 중에서 어느 영역을 교체하여 사용할 것인지를 결정하는 전략

- 페이지 교체 알고리즘

- OPT(OPTimal replacement, 최적 교체) : 앞으로 가장 오랫동안 사용하지 않을 페이지를 교체하는 기법
- FIFO(First In First Out) : 각 페이지가 주기억장치에 적재될 때마다 그때의 시간을 기억시켜 가장 먼저 들어와서 가장 오래 있었던 페이지를 교체하는 기법
- LRU(Least Recently Used) : 최근에 가장 오랫동안 사용하지 않은 페이지를 교체하는 기법

• LFU(Least Frequently Used) : 사용 빈도가 가장 적은 페이지를 교체하는 기법

• NUR(Not Used Recently) : 최근에 사용하지 않은 페이지를 교체하는 기법으로, 참조 비트(Reference Bit)와 변형 비트(Modified Bit)가 사용됨

• SCR(Second Chance Replacement, 2차 기회 교체) : 가장 오랫동안 주기억장치에 있던 페이지 중 자주 사용되는 페이지의 교체를 방지하기 위한 것으로, FIFO 기법의 단점을 보완하는 기법

비선점 Non-preemptive 스케줄링

- 이미 할당된 CPU를 다른 프로세스가 강제로 빼앗아 사용할 수 없는 스케줄링 기법이다.
- FCFS(First Come First Service, 선입 선출) = FIFO(First In First Out) : 준비상태 큐에 도착한 순서에 따라 차례로 CPU를 할당하는 기법으로, 가장 간단한 알고리즘임
- SJF(Shortest Job First, 단기 작업 우선) : 준비상태 큐에서 기다리고 있는 프로세스들 중에서 실행 시간이 가장 짧은 프로세스에게 먼저 CPU를 할당하는 기법
- HRN(Highest Response-ratio Next)
 - 실행 시간이 긴 프로세스에 불리한 SJF 기법을 보완하기 위한 것으로, 대기 시간과 서비스(실행) 시간을 이용하는 기법
 - 우선순위 계산식 : (대기 시간+서비스 시간) / 서비스 시간
- 기한부(Deadline) : 프로세스에게 일정한 시간을 주어 그 시간 안에 프로세스를 완료하도록 하는 기법
- 우선순위(Priority) : 준비상태 큐에서 기다리는 각 프로세스마다 우선순위를 부여하여 그 중 가장 높은 프로세스에게 먼저 CPU를 할당하는 기법

잠깐만요! 에이징(Aging) 기법

시스템에서 특정 프로세스의 우선순위가 낮아 무한정 기다리게 되는 경우, 한 번 양보하거나 기다린 시간에 비례하여 일정 시간이 지나면 우선순위를 한 단계씩 높여 가까운 시간 안에 자원을 할당받도록 하는 기법입니다.

선점 Preemptive 스케줄링

- 하나의 프로세스가 CPU를 할당받아 실행하고 있을 때 우선순위가 높은 다른 프로세스가 CPU를 강제로 빼앗아 사용할 수 있는 스케줄링 기법이다.
- 선점 우선순위 : 준비상태 큐의 프로세스들 중에서 우선순위가 가장 높은 프로세스에게 먼저 CPU를 할당하는 기법
- SRT(Shortest Remaining Time) : 비선점 스케줄링인 SJF 기법을 선점 형태로 변경한 기법으로, 선점 SJF 기법이라고도 함
- 라운드 로빈(RR; Round Robin) : 시분할 시스템(Time Sharing System)을 위해 고안된 방식으로, FCFS 기법과 같이 준비상태 큐에 먼저 들어온 프로세스가 먼저 CPU를 할당받지만 각 프로세스는 시간 할당량(Time Slice, Quantum) 동안만 실행한 후 실행이 완료되지 않으면 다음 프로세스에게 CPU를 넘겨주고 준비상태 큐의 가장 뒤로 배치됨
- 다단계 큐(MQ; Multi-level Queue) : 프로세스를 특정 그룹으로 분류할 수 있을 경우 그룹에 따라 각기 다른 준비상태 큐를 사용하는 기법
- 다단계 피드백 큐(MFQ; Multi-level Feedback Queue) : 특정 그룹의 준비상태 큐에 들어간 프로세스가 다른 준비상태 큐로 이동할 수 없는 다단계 큐 기법을 준비상태 큐 사이를 이동할 수 있도록 개선한 기법

교착상태 Dead Lock

- 상호 배제에 의해 나타나는 문제점으로, 둘 이상의 프로세스들이 자원을 점유한 상태에서 서로 다른 프로세스가 점유하고 있는 자원을 요구하며 무한정 기다리는 현상이다.
- 교착상태 발생의 필요 충분 조건
 - 상호 배제(Mutual Exclusion) : 한 번에 한 개의 프로세스만이 공유 자원을 사용할 수 있어야 함
 - 점유와 대기(Hold and Wait) : 최소한 하나의 자원을 점유하고 있으면서 다른 프로세스에 할당되어 사용되고 있는 자원을 추가로 점유하기 위해 대기하는 프로세스가 있어야 함
 - 비선점(Non-preemption) : 다른 프로세스에 할당된 자원은 사용이 끝날 때까지 강제로 빼앗을 수 없어야 함
 - 환형 대기(Circular Wait) : 공유 자원과 공유 자원을 사용하기 위해 대기하는 프로세스들이 원형으로 구성되어 있어 자신에게 할당된 자원을 점유하면서 앞이나 뒤에 있는 프로세스의 자원을 요구해야 함

교착상태 해결방법

예방 기법 (Prevention)	<ul style="list-style-type: none"> 교착상태가 발생하지 않도록 사전에 시스템을 제어하는 방법으로, 교착상태 발생의 네 가지 조건 중에서 어느 하나를 제거(부정)함으로써 수행됨 종류 : 상호 배제(Mutual Exclusion) 부정, 점유 및 대기(Hold and Wait) 부정, 비선점(Non-preemption) 부정, 환형 대기(Circular Wait) 부정
회피 기법 (Avoidance)	교착상태가 발생할 가능성을 배제하지 않고 교착상태가 발생하면 적절히 피해나가는 방법으로, 주로 은행원 알고리즘(Banker's Algorithm)이 사용됨
발견 (Detection) 기법	시스템에 교착 상태가 발생했는지 점검하여 교착 상태에 있는 프로세스와 자원을 발견하는 것으로, 자원 할당 그래프 등을 사용함
회복 (Recovery) 기법	교착 상태를 일으킨 프로세스를 종료하거나 교착 상태의 프로세스에 할당된 자원을 선점하여 프로세스나 자원을 회복하는 것

프로세스 Process

- PCB를 가지 프로그램
- 실기억장치에 저장된 프로그램
- 프로세서가 할당되는 실체로서, 디스패치가 가능한 단위
- 프로시저가 활동중인 것
- 비동기적 행위를 일으키는 주체
- 지정된 결과를 얻기 위한 일련의 계통적 동작
- 목적 또는 결과에 따라 발생하는 사건들의 과정
- 운영체제가 관리하는 실행 단위

Dispatch

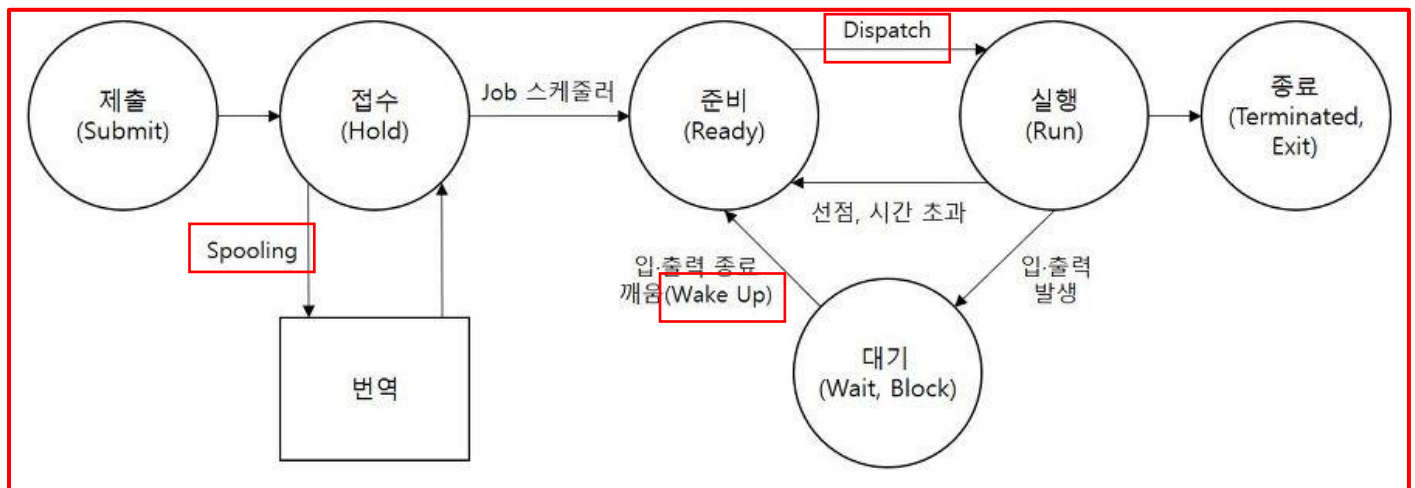
준비 상태에서 대기하고 있는 프로세스 중 하나가 프로세서를 할당받아 실행 상태로 전이되는 과정

Wake up

입출력 작업이 완료되어 프로세스가 대기 상태에서 준비 상태로 전이 되는 과정

Spooling

입출력장치의 공유 및 상대적으로 느린 입출력장치의 처리 속도를 보완하고 다중 프로그래밍 시스템의 성능을 향상시키기 위해 입출력할 데이터를 직접 입출력장치에 보내지 않고 나중에 한꺼번에 입출력하기 위해 디스크에 저장하는 과정



정의 기능	모든 응용 프로그램들이 요구하는 데이터 구조를 지원하기 위해 데이터베이스에 저장될 데이터의 형(Type)과 구조에 대한 정의, 이용방식, 제약 조건등을 명시하는 기능
조작 기능	데이터 검색, 갱신, 삽입, 삭제 등을 체계적으로 처리하기 위해 사용자와 데이터베이스 사이의 인터페이스 수단을 제공하는 기능
제어 기능	데이터베이스를 접근하는 갱신, 삽입, 삭제 작업이 정확하게 수행되어 데이터의 무결성이 유지되도록 제어하는 기능

데이터베이스 Database

특정 조직의 업무를 수행하는 데 필요한 상호 관련된 데이터들의 모임

고급 데이터베이스

통합된 데이터 Integrated Data	검색의 효율성을 위해 중복이 최소화된 데이터의 모임
저장된 데이터 Stored Data	컴퓨터가 접근할 수 있는 저장 매체에 저장된 데이터
운영 데이터 Operational Data	조직의 고유한 업무를 수행하는 데 존재 가치가 확실하고 없어서는 안 될 반드시 필요한 데이터
공용 데이터 Shared Data	여러 응용 시스템들이 공동으로 소유하고 유지하는 데이터

- 데이터 웨어하우스(Data Warehouse) : 급증하는 다량의 데이터를 효과적으로 분석하여 정보화하고 이를 여러 계층의 사용자들이 효율적으로 사용할 수 있도록 한 데이터베이스
- 데이터 마트(Data Mart) : 전사적으로 구축된 데이터 웨어하우스로부터 특정 주제나 부서 중심으로 구축된 소규모 단일 주제의 데이터 웨어하우스를 말함
- 데이터 마이닝(Data Mining) : 데이터 웨어하우스에 저장된 데이터 집합에서 사용자의 요구에 따라 유용하고 가 능성 있는 정보를 발견하기 위한 기법
- OLAP(Online Analytical Processing) : 다차원으로 이루어진 데이터로부터 통계적인 요약 정보를 분석하여 의사 결정에 활용하는 방식
- OLTP(Online Transaction Processing) : 온라인 업무 처리 형태의 하나로 네트워크상의 여러 이용자가 실시간으로 데이터베이스의 데이터를 갱신하거나 검색하는 등의 단위 작업을 처리하는 방식

데이터베이스 관리 시스템 DBMS Database Management System
사용자와 데이터베이스 사이에서 사용자의 요구에 따라 정보를 생성해주고 데이터베이스를 관리해 주는 소프트웨어
- 데이터의 종속성과 중복성의 문제를 해결

정의 기능	모든 응용 프로그램들이 요구하는 데이터 구조를 지원하기 위해 데이터베이스에 저장될 데이터의 형(Type)과 구조에 대한 정의, 이용방식, 제약 조건 등을 명시하는 기능
조작 기능	데이터 검색, 갱신, 삽입, 삭제 등을 체계적으로 처리하기 위해 사용자와 데이터베이스 사이의 인터페이스 수단을 제공하는 기능
제어 기능	데이터베이스를 접근하는 갱신, 삽입, 삭제 작업이 정확하게 수행되어 데이터의 무결성이 유지되도록 제어하는 기능

DBMS는 데이터 구조에 따라 종류 구분

계층형 DBMS	트리 구조를 이용해 데이터의 상호관계를 계층적으로 정의한 DBMS 개체 타입 간에는 상위Owner와 하위Member 관계가 존재하며 일대다(1:N) 대응 관계만 존재	종류 IMS System2000
망형 DBMS	그래프를 이용하여 데이터 논리 구조를 표현한 DBMS 상위와 하위 레코드 사이에서 1:1, 1:N, N:M 대응 모두 지원	IDS TOTAL IDMS
관계형 DBMS	가장 널리 사용되는 DBMS로, 계층형 DBMS와 망형 DBMS의 복잡한 구조를 단순화시킨 DBMS 파일 구조 처럼 구성된 2차원 표table를 하나의 DB로 묶어 테이블 내에 있는 속성들 간의 관계를 설정하거나 테이블 간의 관계를 설정하여 이용 (비관계형으로는 NoSQL)	Oracle SQL Server MySQL

장점	단점
<ul style="list-style-type: none"> • 데이터의 논리적, 물리적 독립성을 보장 • 데이터의 중복을 피할 수 있어 기억 공간 절약 • 저장된 자료를 공동으로 이용 가능 • 데이터의 일관성을 유지 가능 • 데이터의 무결성을 유지 가능 • 보안을 유지 • 데이터를 통전화 할 수 있음 • 데이터를 통합하여 관리 가능 • 항상 최신의 데이터를 유지 • 데이터의 실시간 처리 가능 	<ul style="list-style-type: none"> • 데이터베이스의 전문가 부족 • 전산화 비용이 증가 • 대용량 디스크로의 집중적인 Access로 과부하Overhead가 발생 • 파일의 예비Backup와 회복 Recovery이 어렵 • 시스템이 복잡

ER모형(ERD; Entity Relationship Diagram)

도형	의미
	개체 타입(Entity Type)
	관계 타입(Relationship Type)
	속성
	기본키(Primary Key) 속성
	개체 타입과 속성을 연결
	다중값 속성
	복합 속성
	개체 타입 간의 연관성

개체 Entity

현실 세계의 객체로서 유형 또는 무형의 정보 대상으로 존재하며 서로 구별될 수 있는 것을 뜻하고 개체의 특성을 나타내는 속성을 갖는다.
하나의 개체를 개체 어커런스 Entity Occurance 또는 개체 인스턴스 Entity Instance 라 하고 개체 어커런스의 집합에 대한 공통의 특성들을 갖는 개체 클래스를 개체 타입이라고 한다.

관계 Relationship

2개 이상의 개체 사이에 존재하는 연관성

- 관계에 참여하는 개체 타입의 개수에 대한 차수Degree
n항(n-ary) 관계: 관계에 참여하고 있는 개체 타입이 n개인 관계
- 관계에 참여하는 개체 어커런스의 개수에 대한 대응 카디널리티Mapping Cardinality

1:1 관계	관계에 참여하고 있는 두 개체 타입이 모두 하나씩의 개체 어커런스를 갖는 관계
1: N 관계	관계에 참여하고 있는 개체 타입 중 한 개체 타입은 여러 개의 개체 어커런스를 가질 수 있고, 다른 한 개체 타입은 하나의 개체 어커런스를 갖는 관계
N:M 관계	관계에 참여하고 있는 두 개체 타입 모두 여러 개의 개체 어커런스를 가질 수 있는 관계

속성 attribute

개체의 특성이나 상태를 기술한 것

속성이 가질 수 있는 모든 가능한 값들의 집합을 도메인Domain이라 한다.

- 단순속성 Simple attribute
더 이상 다른 속성으로 나눌 수 없는 속성
- 복합속성 Composite attribute
2개 이상의 속성들로 분해 할 수 있는 속성

관계형 데이터 모델

데이터를 테이블 또는 릴레이션의 구조로 표현하는 논리적 데이터 모델

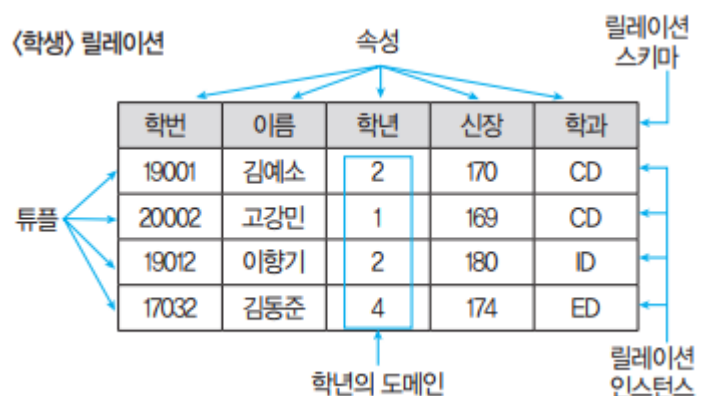
- 관계형 데이터 모델에서는 데이터를 원자 값으로 갖는 이차원의 테이블로 표현하는데 이를 릴레이션이라 한다.
- 릴레이션의 구조는 물리적인 저장 구조를 나타내는 것이 아닌 논리적 구조이므로 다양한 정렬 기준을 통하여 릴레이션을 표현할 수 있다.
- 릴레이션은 구조를 나타내는 릴레이션 스키마와 실제 값들인 릴레이션 인스턴스로 구성
- 릴레이션의 열을 속성이라고 하고 행을 튜플이라고 한다.
- 릴레이션에서 하나의 애트리뷰트가 취할 수 있는 같은 타입의 원자 값들의 집합을 도메인이라 한다.
- 도메인은 실제 애트리뷰트 값이 나타날 때 그 값의 합법 여부를 시스템이 검사하는 데에도 이용된다.
- 개념적 데이터 모델인 ER모델을 논리적 데이터 모델인 릴레이션 스키마로 변환하는 것으로 매핑 룰 Mapping Rule 이라고도 한다.
- ER 도형에서의 개체와 관계는 관계형 데이터 모델에서의 개체 릴레이션과 관계 릴레이션으로 변환하여 표현한다.
- 속성은 컬럼으로, 식별자는 기본키로 표현하고, 릴레이션 간의 관계는 기본키와 이를 참조하는 외래키를 이용하여 표현한다.

개체 A, B와 관계 Y로 이루어진 ER모델을 관계형 데이터 모델에서의 릴레이션 스키마로 변환하는 과정

- 개체 A,B는 각각 독립적인 릴레이션 A,B로 표현된다.
- 각 개체의 속성들은 각 릴레이션의 속성들로 정의
- 기본키는 밑줄을 그어서 표시
- 관계 Y가
- 1:1 관계: 릴레이션 A(or B)의 기본키를 릴레이션 B(or A)의 외래키로 추가
- 1:N 관계: 릴레이션 A의 기본키를 릴레이션 B의 외래키로 추가
- N:M 관계: 릴레이션 A와 B의 기본키를 모두 포함한 별도의 릴레이션으로 표현

이때 생성된 별도의 릴레이션을 교차 릴레이션 Intersection Relation 또는 교차 엔티티 Intersection Entity라고 한다.

기호	의미
	1:1 관계
	1:N 관계
	N:M 관계
	1:0 또는 1:1



키 Key

데이터베이스에서 조건에 만족하는 튜플을 찾거나 순서대로 정렬할 때 기준이 되는 속성

- 슈퍼키(Super Key) : 한 릴레이션 내에 있는 속성들의 집합으로 구성된 키로, 릴레이션을 구성하는 모든 튜플에 대해 유일성(Unique)은 만족하지만, 최소성(Minimality)은 만족하지 못함
- 후보키(Candidate Key) : 릴레이션을 구성하는 속성들 중에서 튜플을 유일하게 식별하기 위해 사용되는 속성들의 부분집합으로, 유일성과 최소성을 모두 만족함

- 기본키(Primary Key) : 후보키 중에서 특별히 선정된 키로 중복된 값과 NULL 값을 가질 수 없음
- 대체키(Alternate Key) : 후보키 중에서 선정된 기본키를 제외한 나머지 후보키를 의미함
- 외래키(Foreign Key) : 다른 릴레이션의 기본키를 참조하는 속성 또는 속성들의 집합을 의미하며, 릴레이션 간의 관계를 표현할 때 사용함

무결성 Integrity

데이터베이스에 저장된 데이터 값과 그것이 표현하는 현실 세계의 실제값이 일치하는 정확성을 의미

무결성 제약 조건

데이터베이스에 들어 있는 데이터의 정확성을 보장하기 위해 부정확한 자료가 데이터베이스 내에 저장되는 것을 방지하기 위한 제약조건

- 개체 무결성(Entity Integrity, 실체 무결성) : 기본 테이블의 기본키를 구성하는 어떤 속성도 Null 값이나 중복값을 가질 수 없다는 규정
- 도메인 무결성(Domain Integrity, 영역 무결성) : 주어진 속성 값이 정의된 도메인에 속한 값이어야 한다는 규정
- 참조 무결성(Referential Integrity) : 외래키 값은 Null이거나 참조 릴레이션의 기본키 값과 동일해야 함. 즉 릴레이션은 참조할 수 없는 외래키 값을 가질 수 없다는 규정
- 사용자 정의 무결성(User-Defined Integrity) : 속성 값들이 사용자가 정의한 제약조건에 만족해야 한다는 규정
- NULL 무결성 : 릴레이션의 특정 속성 값이 NULL이 될 수 없도록 하는 규정
- 고유(Unique) 무결성 : 릴레이션의 특정 속성에 대해 각 튜플이 갖는 속성값들이 서로 달라야 한다는 규정
- 키(Key) 무결성 : 하나의 릴레이션에는 적어도 하나의 키가 존재해야 한다는 규정
- 관계(Relationship) 무결성 : 릴레이션에 어느 한 튜플의 삽입 가능 여부 또는 한 릴레이션과 다른 릴레이션의 튜플들 사이의 관계에 대한 적절성 여부를 지정한 규정

네트워크 Network

두 대 이상의 컴퓨터를 전화선이나 케이블 등으로 연결하여 자원을 공유하는 것

근거리 통신망 LAN Local Area Network	비교적 가까운 거리에 있는 컴퓨터, 프린터 등과 같은 자원에 연결하여 구성 주로 자원 공유를 목적으로 사용 사이트 간의 거리가 짧아 데이터의 전송 속도가 빠르고 에러 발생률이 낮다. 주로 버스형이나 링형 구조를 사용
광대역 통신망 WAN Wide Area Network	멀리 떨어진 사이트들을 연결하여 구성 사이트 간의 거리가 멀기 때문에 통신 속도가 느리고 에러 발생률이 높다. 일정한 지역에 있는 사이트들을 근거리 통신망으로 연결한 후 각 근거리 통신망을 연결하는 방식

인터넷 Internet

TCP/IP 프로토콜을 기반으로 하여 전 세계 수많은 컴퓨터와 네트워크들이 연결된 광범위한 컴퓨터 통신망

1. UNIX 운영체제 기반
2. 고유한 IP주소 보유
3. 연결하기 위해서 브리지, 라우터, 게이트웨이를 사용
4. 백본 Backbone: 다른 네트워크 또는 같은 네트워크를 연결하기 위해 중추적 역할을 하는 네트워크

주로 인터넷의 주가 되는 기간망을 일컫는 용어

IP 주소 Internet Protocol Address

인터넷에 연결된 모든 컴퓨터 자원을 구분하기 위한 고유한 주소

8비트씩 4부분, 총 32비트로 구성

네트워크 부분의 길이에 따라 5단계로 구분

(A class: 국가나 대형 통신망~중대형, 소규모, 멀티캐스트용~E: 실험적 주소, 공용되지 않음)

IPv6 Internet Protocol version 6

- 현재 사용하고 있는 IP 주소 체계인 IPv4의 주소 부족 문제를 해결하기 위해 개발되었다.
 - 16비트씩 8부분, 총 128비트로 구성되어 있다.
 - 각 부분을 16진수로 표현하고, 콜론(:)으로 구분한다.
 - IPv4에 비해 자료 전송 속도가 빠르고, IPv4와 호환성이 뛰어나다.
 - 인증성, 기밀성, 데이터 무결성의 지원으로 보안 문제를 해결할 수 있다.
 - IPv6의 주소 체계
 - 유니캐스트(Unicast) : 단일 송신자와 단일 수신자 간의 통신(1:1 통신에 사용)
 - 멀티캐스트(Multicast) : 단일 송신자와 다중 수신자 간의 통신(1:N 통신에 사용)
 - 애니캐스트(Anycast) : 단일 송신자와 가장 가까이 있는 단일 수신자 간의 통신(1:1 통신에 사용)
 - 도메인 네임 : 숫자로 된 IP 주소를 사람이 이해하기 쉬운 문자 형태로 표현한 것으로, 문자로 된 도메인 네임을 컴퓨터가 이해할 수 있는 IP 주소로 변환하는 역할을 하는 시스템을 DNS(Domain Name System)라고 함
- 기타
- TCP/IP : 인터넷에 연결된 서로 다른 기종의 컴퓨터들이 데이터를 주고받을 수 있도록 하는 표준 프로토콜

PDU	NIC	OSI	TCP/IP	주요 프로토콜
메시지		응용 계층 (서비스 제공)	응용계층 (응용 프로그램 간의 데이터 송수신 제공)	FTP, SMTP, TELNET SNMP, DNS, HTTP
		표현계층 (변환)		
		세션 계층 (대화제어, 동기점 synchronization point)		
세그먼트	게이트웨이 (프로토콜이 다른 네트워크에 연결, 출입구 역할)	전송 계층 (인터페이스, 상위 ↑ 단말기 사이에~ End- to end 종단 시스템)	전송계층 (호스트들 간의 신뢰성 있는 통신을 제공)	TCP UDP RTCP
패킷	라우터 Router (서로 다른 LAN 연결, 경로 설정)	하위 ↓ 네트워크 계층 (데이터 교환, 경로 설정)	인터넷 계층 (데이터 전송을 위한 주소 지정, 경로 설정을 제공)	IP ICMP IGMP ARP RARP
프레임	랜카드 브리지 (LAN, MAC) 스위치	데이터 링크 계층 (흐름제어, 프라임동기화 기능, 오류제어, 순서제어)	네트워크 액세스 계층 (실제 데이터(프레임)를 송수신하는 역할)	Ethernet IEEE HDLC X.25 RS-232C
비트	리피터 (신호재생) 허브 (가까운 거리, 대부분 스위치 허브)	물리 계층 (절차적, 기계적, 전기적, 기능적)		

프로토콜 데이터 단위 PDU Protocol Data Unit

네트워크 인터페이스 카드 NIC Network Interface Card

- 네트워크 관련 장비

• 허브(Hub)

- 한 사무실이나 가까운 거리의 컴퓨터들을 연결하는 장치로, 각 회선을 통합적으로 관리하며, 신호 증폭 기능을 하는 리피터의 역할도 포함함
- 더미 허브(Dummy Hub) : 네트워크에 흐르는 모든 데이터를 단순히 연결하는 기능만을 제공함
- 스위칭 허브(Switching Hub) : 네트워크상에 흐르는 데이터의 유무 및 흐름을 제어하여 각각의 노드가 허브의 최대 대역폭을 사용할 수 있는 지능형 허브임

- 리피터(Repeater) : 물리 계층의 장비로, 전송되는 신호가 왜곡되거나 약해질 경우 원래의 신호 형태로 재생함
- 브리지(Bridge) : 데이터 링크 계층의 장비로, LAN과 LAN을 연결하거나 LAN 안에서의 컴퓨터 그룹을 연결함
- 라우터(Router) : 네트워크 계층의 장비로, LAN과 LAN의 연결 및 경로 선택, 서로 다른 LAN이나 LAN과 WAN을 연결함
- 게이트웨이(Gateway) : 전 계층(1~7계층)의 프로토콜 구조가 전혀 다른 네트워크의 연결을 수행함
- 스위치(Switch) : 브리지와 같이 LAN과 LAN을 연결하여 훨씬 더 큰 LAN을 만드는 장치

L2, L3, L4, L7 스위치

OSI Open System Interconnection

TCP/IP

인터넷에 연결된 서로 다른 기종의 컴퓨터들이 데이터를 주고 받을 수 있도록 하는 표준 프로토콜

프로토콜	TCP	IP
연결성	연결형 서비스	비연결형 서비스
신뢰성	높음	낮음
속도	느림	빠름
패킷 교환 방식	가상 회선 방식	데이터그램 방식
수신 여부	수신함	수신 안 함

프로토콜

서로 다른 기기들 간의 데이터 교환을 원활하게 수행할 수 있도록 표준화시켜 놓은 통신 규약

- 기본요소

구문(Syntax)	전송하고자 하는 데이터의 형식, 부호화, 신호 레벨 등을 규정함
의미(Semantics)	두 기기 간의 효율적이고 정확한 정보 전송을 위한 협조 사항과 오류 관리를 위한 제어 정보를 규정함
시간(Timing)	두 기기 간의 통신 속도, 메시지의 순서 제어 등을 규정함

응용계층 프로토콜

- FTP(File Transfer Protocol) : 컴퓨터와 컴퓨터 또는 컴퓨터와 인터넷 사이에서 파일을 주고받을 수 있도록 하는 원격 파일 전송 프로토콜
- SMTP(Simple Mail Transfer Protocol) : 전자 우편을 교환하는 서비스
- TELNET : 멀리 떨어져 있는 컴퓨터에 접속하여 자신의 컴퓨터처럼 사용할 수 있도록 해주는 서비스
- SNMP(Simple Network Management Protocol) : TCP/IP의 네트워크 관리 프로토콜로, 라우터나 허브 등 네트워크 기기의 네트워크 정보를 네트워크 관리 시스템에 보내는 데 사용되는 표준 통신 규약
- DNS(Domain Name System) : 도메인 이름을 IP 주소로 매핑(Mapping)하는 시스템
- HTTP(HyperText Transfer Protocol) : 월드 와이드 웹(WWW)에서 HTML 문서를 송수신 하기 위한 표준 프로토콜

전송계층 프로토콜

- TCP(Transmission Control Protocol)
 - 양방향 연결(Full Duplex Connection)형 서비스를 제공한다.
 - 가상 회선 연결(Virtual Circuit Connection) 형태의 서비스를 제공한다.
 - 스트림 위주의 전달(패킷 단위)을 한다.
 - 신뢰성 있는 경로를 확립하고 메시지 전송을 감독한다.
- UDP(User Datagram Protocol)
 - 데이터 전송 전에 연결을 설정하지 않는 비연결형 서비스를 제공한다.
 - TCP에 비해 상대적으로 단순한 헤더 구조를 가지므로, 오버헤드가 적다.
 - 고속의 안정성 있는 전송 매체를 사용하여 빠른 속도를 필요로 하는 경우, 동시에 여러 사용자에게 데이터를 전달할 경우, 정기적으로 반복해서 전송할 경우에 사용한다.
 - 실시간 전송에 유리하며, 신뢰성보다는 속도가 중요시되는 네트워크에서 사용된다.
- RTPC(Real-Time Control Protocol)
 - RTP(Real-time Transport Protocol) 패킷의 전송 품질을 제어하기 위한 제어 프로토콜이다.
 - 세션(Session)에 참여한 각 참여자들에게 주기적으로 제어 정보를 전송한다.
 - 하위 프로토콜은 데이터 패킷과 제어 패킷의 다중화(Multiplexing)를 제공한다.

인터넷계층 프로토콜

- IP(Internet Protocol) : 전송할 데이터에 주소 지정 및 경로 설정 등의 기능을 하며, 비연결형인 데이터그램 방식을 사용하므로 신뢰성이 보장되지 않음
- ICMP(Internet Control Message Protocol) : IP와 조합하여 통신중에 발생하는 오류의 처리와 전송 경로 변경 등을 위한 제어 메시지를 관리하는 역할을 하며, 헤더는 8Byte로 구성됨
- IGMP(Internet Group Management Protocol) : 멀티캐스트를 지원하는 호스트나 라우터 사이에서 멀티캐스트 그룹 유지를 위해 사용됨
- ARP(Address Resolution Protocol) : 호스트의 IP 주소를 호스트와 연결된 네트워크 접속 장치의 물리적 주소(MAC Address)로 바꿈
- RARP(Reverse Address Resolution Protocol) : ARP와 반대로 물리적 주소를 IP 주소로 변환하는 기능을 함

데이터 교환 방식에는 회선 교환 방식과 축적 교환 방식이 있으며

축적 교환 방식에는 메시지 교환 방식과 패킷 교환 방식이 있다.

회선 교환 방식 Circuit Switching

통신을 원하는 두 지점을 교환기를 이용하여 물리적으로 접속시키는 방식

- 공간 분할 교환 방식 SDS Space Division Switching (기계식 접점과 전자 교환기의 전자식 접점 등을 이용)
- 시분할 교환 방식 TDS Time Division Switching (전자 부품이 갖는 고속성과 디지털 교환 기술을 이용)

패킷 교환 방식 Packet Switching

메시지를 일정한 길이의 패킷으로 잘라서 전송하는 방식

OSI 7계층의 네트워크 계층에 해당

접속 규정 X.25 프로토콜 X.75 권고안 X.3

라우팅(경로제어 Routing)

송수신 측 간의 전송 경로 중에서 최적 패킷 교환 경로를 결정하는 기능

경로제어표 Routing Table를 참조하여 라우터에 의해 수행

- 라우팅의 요소

성능기준, 경로의 결정 시간과 장소, 정보 발생지, 경로 정보의 갱신 시간

- RIP(Routing Information Protocol) : 현재 가장 널리 사용되는 라우팅 프로토콜로, 소규모 동종의 네트워크 내에서 효율적인 방법이며, 최대 홉수를 15로 제한함
- IGRP(Interior Gateway Routing Protocol) : RIP의 단점을 보완하기 위해 만들어 개발된 것으로, 네트워크 상태를 고려하여 라우팅하며, 중규모 네트워크에 적합함
- OSPF(Open Shortest Path First Protocol) : 대규모 네트워크에서 많이 사용되는 라우팅 프로토콜로, 라우팅 정보에 변화가 생길 경우 변화된 정보만 네트워크 내의 모든 라우터에 알리며, RIP에 비해 홉수에 제한이 없음
- BGP(Border Gateway Protocol) : 자율 시스템(AS) 간의 라우팅 프로토콜로, EGP의 단점을 보완하기 위해 개발되었음

라우팅 알고리즘

데이터가 송신측에서 수신측까지 전송되는 과정에서는 많은 물리적 장치들을 거치는데 그 과정에서 목적지 까지 도착하기 위한 최적 경로를 산출하기 위한 알고리즘

- 거리 벡터 알고리즘 Distance Vector Algorithm : RIP IGRP
- 링크 상태 알고리즘 Link State Algorithm : OSPF