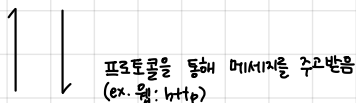


1. 클라이언트와 서버

- Web server, File server (File 공유 사이트 | ex) 디스크 등
- 웹 페이지의 자원, 공유 데이터의 처리 및 저장 등의 비즈니스 로직 수행
- 사용자와의 거리 또한 속도에 영향을 준다.



- 웹 브라우저 : 웹서버로 접속하기 위한 터미널
- 서버에 자원을 요청하고, 서버가 주는 응답을 제공 받는다.
- 주로 사물과 밀접 수월
- P2P, 브로케인 등은 클라이언트 일과 동시에 서버의 역할도 수행한다.

2) 어떻게 하나? (OP-Operation System) 컴퓨터 시스템의 자원을 효율적으로 관리, 사용자가 컴퓨터를 편리하게 이용할 수 있도록 환경을 제공하는 여러 프로그램의 모임 처리능력, 사용자호출, 속도 향상, 반환시간 단축이 이들 중 하나이다.

- 2) 종류 - 단일 작업처리 시스템: DOS
- 다중 " " : WINDOWS. UNIX. LINUX

- 3) 기술
 - 프로그래밍, 기하학, ^{이동}입출력, 파일 및 자료들의 지향관
 - 효율적인 연구를 위해 자료의 설계/구현
 - 섀들라-시드탈만 인턴파괴의 계몽
 - 데이터의 난해 및 응용
 - 시드탈만 연구 검사 불가
 - 입출력, 구조
 - 시드탈만 각종 응용데이터와 네트웍을 연구

1) GUI : 사용자가 편하게 이용할 수 있도록 컴퓨터의 기능을 아이콘이나 그래픽, 문자 알기쉽게 나타낸 것으로, 어떤 행동을 제어하기 위한 그래픽을 적절 조합하여야함 (ex. ☒ 선택식 삭제, 저장 등의 기능) MP3, 스마트폰, 스마트 가전 등 많은 휴대 장치에 보편 적용이 필수

- 2) TUI : 문화의 향유를 권유함 - 생활가 입문하는 젊은층을 향하여, 인문과학, 사회과학 및 공예·디자인 분야까지 상대적으로 저렴한 가격에

- 3) Port (포트): 컴퓨터의 출구 (출신번호)
- 4) Route (라우터): 데이터가 목적지까지 가는 경로를 안내한다
↳ Routing: 네트워크를 통해 데이터를 보내는 과정의 전체를 의미하는 과정
- 5) DNS (Domain Name System): 사람이 읽을 수 있는 도메인 명 (ex. www,com)을 기계가 읽을 수 있는 IP주소로 변환하는 시스템
- 6) NAT (Network Address Translation): 1. 외부로 연결된 공인 (Public) IP와 내망 네트워크의 사설 (private) IP가 다르면, 내부호출을 위해서 두 IP를 매핑해서 변환할 수 있어야 함. (비공인 주소인)
- 7) 랜트 풀업임 : (포드 매핑) 컴퓨터 네트워킹에서 게이트웨이를 가리키는 용어. 하나의 IP와 모든 포트는 정형화되어있음. 어느곳으로 연결할지 NAT 가해서 있음

- 1) SSH (Secure Shell Protocol): 컴퓨터간의 인터넷 망을 통해 데이터를 전송할 때 보안상 안전하게 통신하게끔 하는 프로토콜
ex. 데이터 전송, 원격제어
- 2) Putting : SSH, 통신을 위한 클라이언트와 서버, 암호화 모듈, 인증서, 프로토콜, 터미널, 유닉스 시스템, 데이터베이스 (database)
- 3) Port : 컴퓨터의 포트 (통신구)

1) Web server : 클라이언트가 서버에 페이지 요청하면 요청을 받아 가장 앞단에서 응답하여 정적 콘텐츠 (이미지, 파일 등)를 클라이언트 서버
대응 : Apache 유저 : 지역적 영향 X

2) Web Application Server : 동적 콘텐츠 제공을 위해 만들어진 애플리케이션 서버 (DB 조회 로직 처리 등)
응답 : tomcat 유저 : 지역적 영향 유 (예 : 네트워킹과 응답의 많음/적음)

3) 동적 콘텐츠 처리를 하는가? → Y : Web App 동적 콘텐츠 모두 가능하지만 정적 콘텐츠 처리가 속 영향
→ N : Web Server

클라이언트 \longleftrightarrow 서버

1) 사용자에게 필요한 부분
ex) 웹서버 : 클라이언트에서 받은 요청을 처리하고 결과를 응답하여 출력.
ex) 웹동. 네이비 등

→ 대부분 "클라이언트 서버" 구조로 되어 있는데,
이 중 우리가 배울 리눅스(LINUX)는 서버의 운영체제

* 운영체제 : 컴퓨터의 HW와 SW를 제어하여, 사용자가 컴퓨터를 쓸 수 있게 만드는 프로그램

- 2) 운영체제에는 우리가 익히 아는 WINDOWS도 있지만 GUI 기반이라 서버에서는 불필요한
변종이 많아 TUI 기반의 LINUX / UNIX 이용비율이 90% 이상이다.

- 3) UNIX는 LINUX가 개발되기 전 많이 쓰이던 서버 운영체제로 안정성과 대용량 처리 능력이 뛰어났다.

LINUX는 UNIX와 달리 오픈소스(OPEN SOURCE)라 접근성이 좋으며, 최근 개인 PC, 소형기전, 모바일 등 다양한 분야에 이용되며 그 영역을 넓혀가고 있다.

- 4) LINUX 메이커를 크게 ^{FREE} Ubuntu 인 ^{CHARGE} CentOS 가 있음. 둘이 서로 많이 다른 느낌.
오픈소스 지루하는 Ubuntu를 수업에서 이용

- 5) 실제 서버는 혼자 실행하는 환경에 부적합. Virtual Box를 통해 Ubuntu를 실행할 수 있는 가상 서버를 만들어 수월함

* 금일 수업 시간에 이용한 LINUX 명령어.

- 1) su - : 일반 사용자 → root 권한 전환
root @도미이명 : ~#
↳ ~ : 홈디렉토리 / # : 일반 사용자 / \$: 관리자
- 2) sudo : 한글영문표기, root 권한을 일반 사용자 share권 부여해주는 .
- 3) apt : 표준인 패키지 프로그램을 설치할 때, root 권한 필요.
ex) apt clean : 저장된 PKG 정보 초기화
apt update : 패키지 정보를 최신 상태로 갱신 (install X, 유닉스 최신화)
apt install @ ssh, apache2, tomcat9
(필수) (선택) (DB)
mysql-server mysql-client
- 4) clear : 화면 정보를 지움
- 5) Sys 종료, 재부팅
→ 종료 : shutdown now : 각 프로세스 다 닫고 종료 시킴
→ reboot : shutdown -r now

1. 강의 들어가기

학습내용 소개

- 리눅스는 기업이나 공공기관, 학교 등 **정보시스템 서비스를 제공하는 서버의 운영체제**로 많이 사용되는 소프트웨어이다. 운영체제
- 리눅스 운영체제를 개인용 PC의 윈도우 운영체제를 대신하는 운영체제로 사용할 수도 있다. 서버, 개인사용자, 크롬 기기에서도. 즉 사용되는 맥도!
- 또한 스마트폰에서 많이 사용되는 **안드로이드 운영체제**에 기본 바탕은 리눅스로 구성되어 있다. 리눅스
- 본 과목에서는 **서버형식의 리눅스 운영체제를 사용하는 방법 및 이를 활용하여 웹 프로그램을 다루는 방법**을 배우고 실습한다.
- 이를 위하여 본 단위에서는 운영체제, 서버 등 관련용어를 정리하고, 리눅스 운영체제를 설치하고 이 시스템에 접속하여 간단한 활용을 하는 부분은 기본이며, 이를 활용한 개발환경을 구축한다.

학습목표 제시

- 유닉스와 리눅스라는 OS를 정의할 수 있다.
- 일반적인 리눅스의 종류인 데스크 탑 형식과 본 과목에서 학습하는 서버형식에 대하여 설명할 수 있다.
- 실습학습 중 어렵거나 혼자서 수행하기 어려울 때 게시판 등 커뮤니티 활용 및 웹사이트 검색 등 도움되는 정보와 문제해결방법을 고안할 수 있다.
- 클라이언트/서버환경의 개발자 환경을 스스로 구축할 수 있다.

iamhpd@kopo.ac.kr

2



3. 이해하기

1. 클라이언트 서버 환경

1) 개요

- 일상적 정보처리 시스템은 클라이언트/서버 환경으로 구성되어 있음(vs Standalone)
- 클라이언트는 사용자에게 보여지는 부분
- 서버는 클라이언트가 수행할 수 있는 서비스를 지원하는 부분



iamhpd@kopo.ac.kr

4

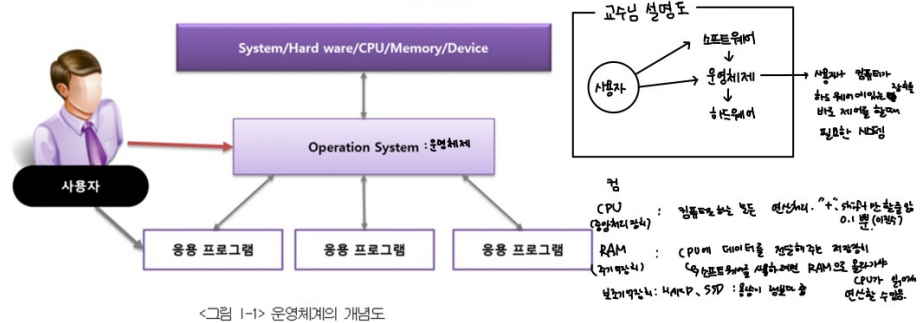


- 3. 이해하기

1. 클라이언트 서버 환경

2) 서버의 운영체제

- 클라이언트는 화면에 보이는 것, 기기의 다양한 기능 사용, 사용자 중심
- 서버는 데이터처리, 통신, 업무 중심, 대용량, 가용성, 확장성, 안정성
- 서버의 운영체제는? Windows보다는 linux/Unix가 유리하며 90%이상을 차지



보충학습

운영체제는 하드웨어와 소프트웨어의 중간에 위치

사용자가 컴퓨터 시스템을 사용하기 위하여 운영체제는 사용자의 명령을 하드웨어에 전달하고, 응용프로그램이 하드웨어 자원 등을 사용하는 경우, 하드웨어 제어 운영을 중간에서 조정하고 관리하는 역할을 수행하고 있다. 운영체제는 사용자와 응용프로그램이 복잡한 컴퓨터 하드웨어 부분을 이해하지 않아도 쉽게 사용되거나 운영될 수 있도록 해주는 시스템 프로그램이다.

● 3. 이해하기

1. 클라이언트 서버 환경

2) 서버의 운영체제

- 클라이언트는 화면에 보이는 것, 기기의 다양한 기능 사용, 사용자 중심
- 서버는 데이터처리, 통신, 업무 중심, 대용량, 가용성, 확장성, 안정성
- 서버의 운영체제는? Windows보다는 linux/Unix가 유리하며 90%이상을 차지
- 앞으로 교안에 나오는 그림 뜯어 붙인 내용은 교재 내용임.

1) 운영체제(Operating System, OS)

① 운영체제 : 컴퓨터의 하드웨어와 소프트웨어를 제어하여, 사용자가 컴퓨터를 쓸 수 있게 만들어주는 프로그램. <그림 1-1>

② 운영체계의 기능.

- 하드웨어와 응용프로그램간의 인터페이스 역할 하드웨어를 편하게 잘 보라할 수 있는 인터페이스
- CPU, 주기억장치, 입출력장치 등의 컴퓨터 자원 관리 ↳ 소프트웨어 개발자가 하드웨어에 대해 잘 알 필요가 없음
- 인간과 컴퓨터간의 상호작용을 제공함과 동시에 컴퓨터의 동작을 구동(Booting)
- 작업의 순서를 정하며 입출력 연산을 제어
- 프로그램의 실행을 제어
- 데이터와 파일의 저장을 관리

3. 이해하기

1. 클라이언트 서버 환경

4) 데스크탑 형식과 서버형식

- 리눅스 : 서버의 OS역할 뿐만 아니라, 현재 윈도우(MacOS)의 PC OS의 역할도 담당할뿐..
- 즉 text기반 OS에서 Windows같은 UI기반의 데스크탑 형식에 대한 노력
- 안드로이드의 탄생 -> 리눅스 데스크탑이 진화한 모바일 버전
- 많은 리눅스 메이커가 존재(명령어들은 대부분 비슷)
- 시장강자 Ubuntu (70~80%), 또다른 강자 CentOS (나머지) 원래 CentOS랑 많이 맞음
- Ubuntu : 시장에 강자다 보니 참고할 문서, 구글링이 쉽다. 오픈소스계열 SMB의 강자
- CentOS : 돈주고 사는 기존 기업용 리눅스인 red Hat과 호환, 오라클 사용 가능, 큰 기업 고려

- Ubuntu는 데스크탑, 모바일, **서버**, 클라우드등의 다양한 버전을 제공

→ 수업 중 이용

- 데스크탑과 달리 서버형식은 딱히 화면을 보여주는 윈도우UI가 필요하지 않다.
- 경량화, 고성능화를 위하여 굳이 낭비할 필요는 없다(근데 관리자도 UI편한게 더 좋은데..)

WINDOW : GUI기반 운영체제 : 서버 입장에서선 비효율적
LINUX : TUI기반 (text)

iamhpd@kopo.ac.kr

8



4. 실습하기(1)

4. 몇 가지 리눅스 명령어를 급히 익히자(리눅스 배우지도 안았는데.. 일단 그냥 무따기, 나중에 또 복습함)

1) su - : 일반 사용자에서 root 사용자 (관리자)로 로그인 전환

2) sudo : 한줄 명령어로 root권한 잠시 사용 : Install할때에 권한 부여.

ex) sudo passwd root : 처음 일반 사용자로 접속 후 root의 패스워드가 설정이 안되어 있는 경우 root사용자의 패스워드를 바꾸는 명령

3) apt : 필요한 패키지 프로그램을 설치하는 명령으로 반드시 root 권한으로 사용

ex) apt clean : 저장된 패키지정보 (설치할 프로그램을 다운로드 받을 수 있는 웹페이지 주소를 가지고 있는 정보) 를 초기화

apt update : 패키지 정보를 최신상태로 갱신

✓ apt install ssh : [ssh] 를 설치 // 서버 원격접속 서비스

✓ apt install apache2 : [아파치 웹서버] 를 설치 //웹서버(문서) 서비스

apt install tomcat9 : [톰캣]을 설치 // 웹서버(프로그램실행) 서비스

apt install mysql-server mysql-client : [mysql]을 설치 //데이터베이스

4) 시스템 종료와 재부팅

- 시스템 종료 : shutdown now 명령 이후 전 프로세스를 다 내린 후 전원 off

- 재 시작 : shutdown -r now

iamhpd@kopo.ac.kr



4. 실습하기(1)

5. 네트워크 설정

1) DHCP : 네트워크 IP를 접속 시마다 임의로 배정하는 유동IP방식

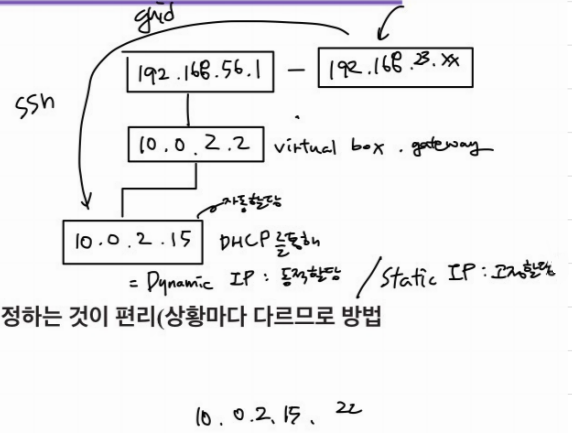
2) PC에서 배정된 IP확인 : cmd명령창 -> ipconfig

- 만일 공유기를 사용한다면 사용하는 PC의 IP가 고정적으로 배정되도록 설정하는 것이 편리(상황마다 다르므로 방법은 검색이나 공유기 사용설명서 등을 참고)

포트 매핑 : 포트 포워딩

3) 서버에서 배정된 IP확인 : ifconfig -a

- 다음 장 계속-



iamhpd@kopo.ac.kr



< 진도 외 수업 내용 >

웹 최신 개발 트렌드

백엔드 - Java : Spring, Jsp...
Backend (서버) - Python : Django, Flask...
- Javascript (NodeJs) : Express..
- php, asp ... Ruby on Rails

프론트엔드 - HTML Hyper Text Markup Language (XML유형 ex. < > </ >)

Frontend (클라이언트) - CSS : Styler (Good look)

- Javascript Hot Iquery

less, Scss : transfile

Framework { Angular 같은 프레임워크 개발의 편리, 손쉬움
React : 1st sit < one source multi-view
VUE

Back - Front 나뉘게 된 계기
웹 자원 접속시 느린 속도, 다양한 것 표준화하여 쓰기
Rest API
v AJAX (웹의 발전) : 페이지가 안 바뀌면서 서버와 소통
v SPA : Single Page Application

IP란? 0~255까지의 숫자만 쓸수 있음.
192.168.23.34
8bit 8bit 8bit 8bit

컴 0~1 2진법
8bit

IP => IPV4

IP ver.4 32 bit 주소체계 사용 : 전 세계에 40억개만 있음

$$\begin{aligned} 2^{32} &= 2^8 \times 2^8 \times 2^8 \times 2^8 \\ &= 2^8 \times (1024)^3 \\ &= 2^8 \times 10^9 \approx 40억 \end{aligned}$$

IP : 192.168.23.34/24
Netmask : 255.255.255.0
Gateway : 192.168.23.1

