

운영체제 기초 활용하기

컴퓨터 하드웨어의 구성

컴퓨터 하드웨어의 구성

■ 컴퓨터 시스템

- 데이터를 처리하는 물리적인 기계장치인 하드웨어^{hardware}와 어떤 작업을 지시하는 명령어로 작성한 프로그램인 소프트웨어^{software}로 구성

■ 컴퓨터 하드웨어

- 하드웨어는 프로세서, 메모리(기억장치), 주변장치로 구성되고, 이들은 시스템 버스로 연결

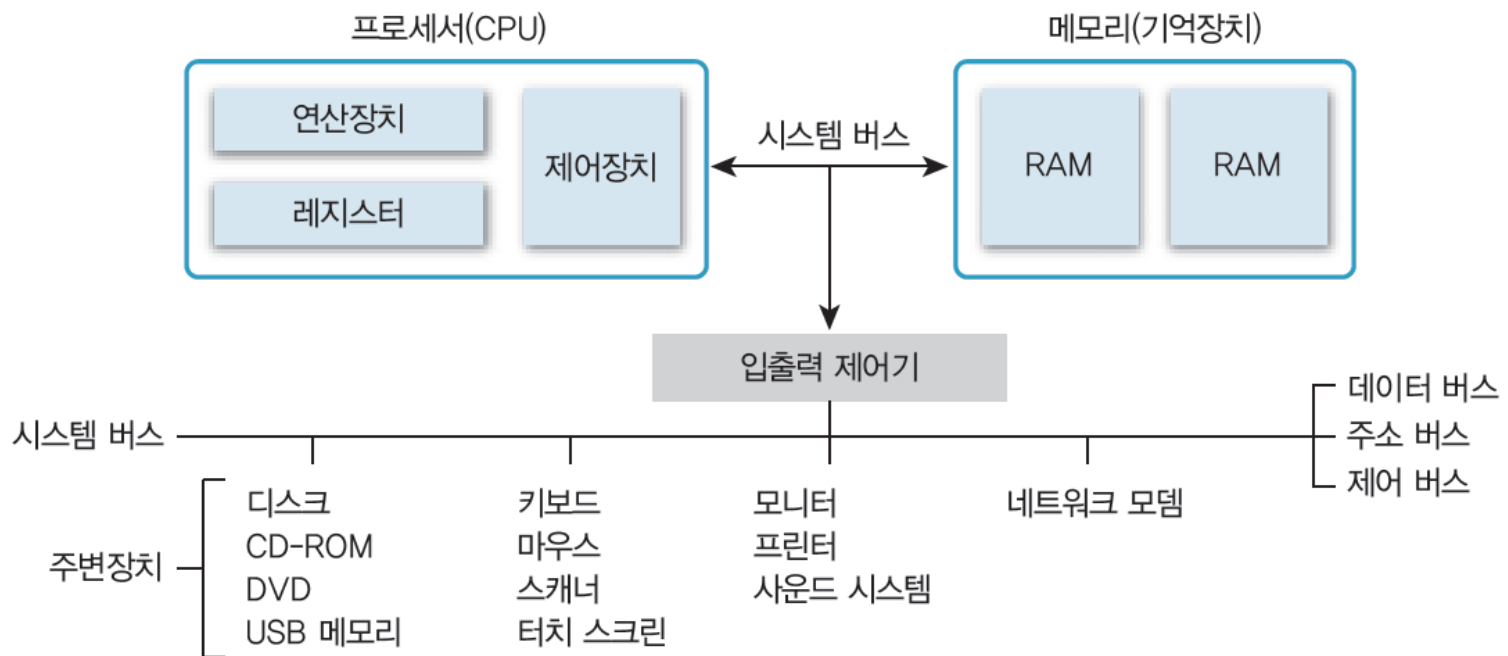


그림 1-1 컴퓨터 하드웨어의 구성

운영체제 (Operation System)

운영체제의 개념과 발전 목적

■ 운영체제의 개념

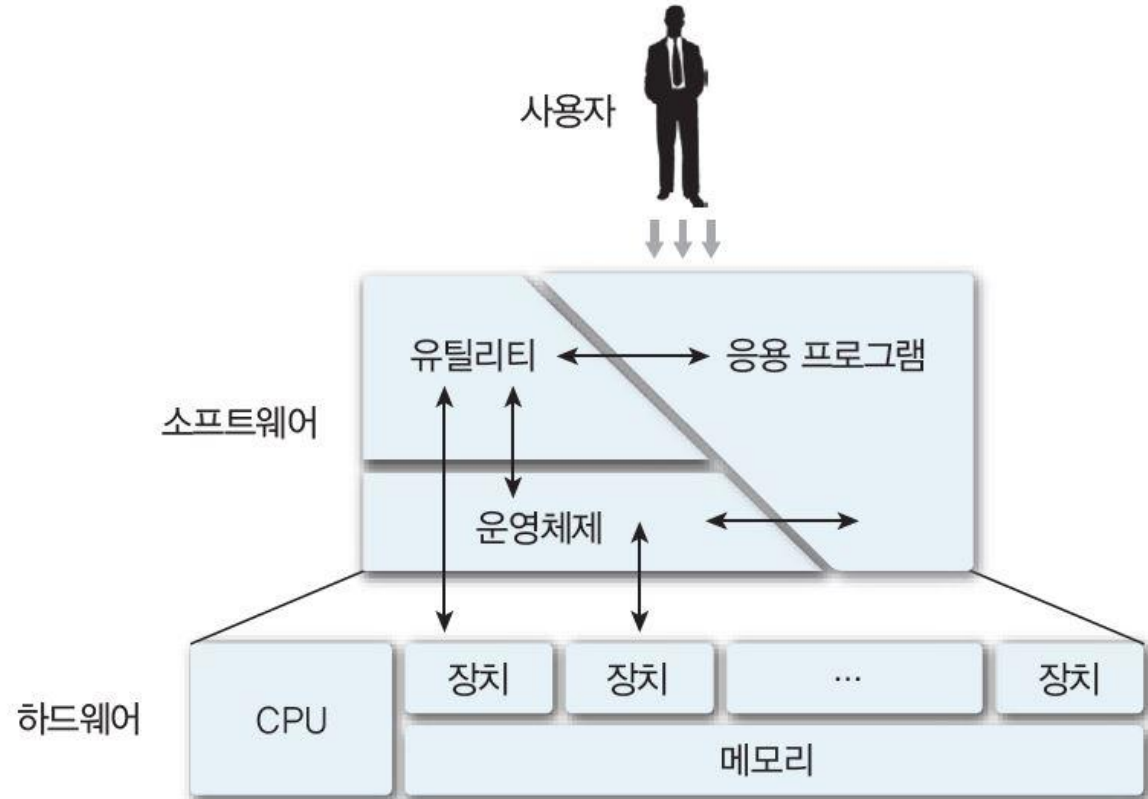
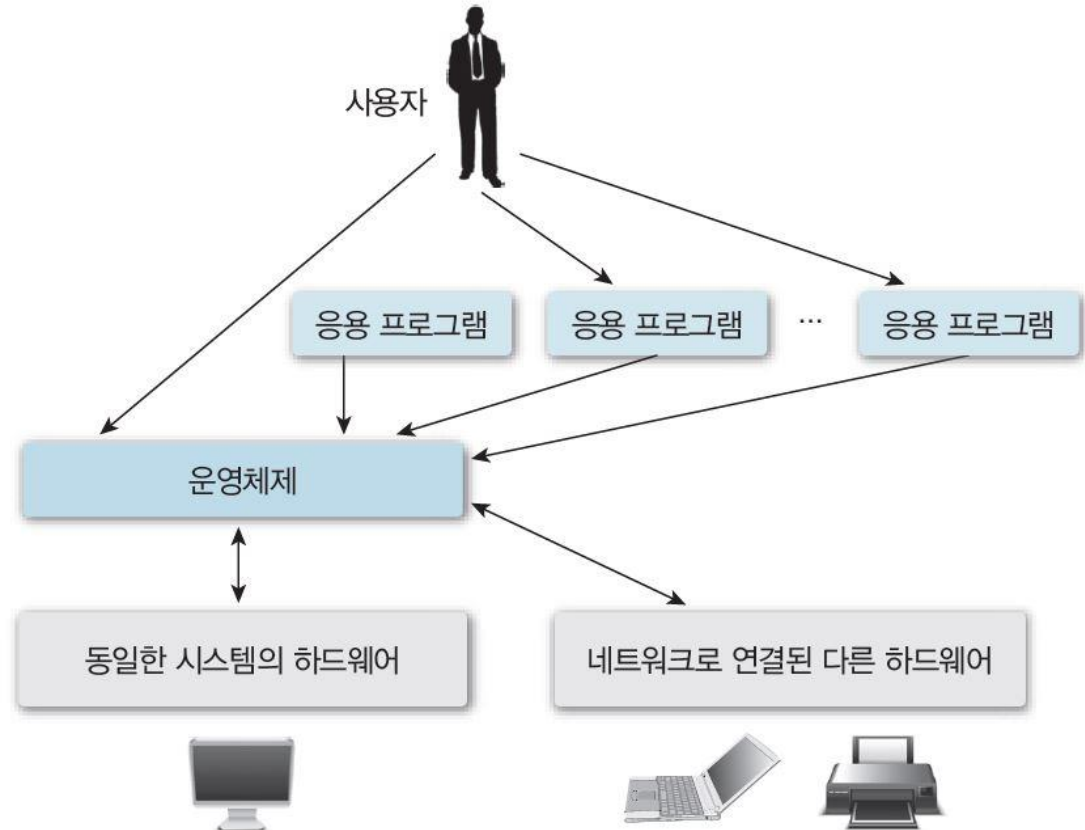


그림 2-1 컴퓨터 시스템의 구성 요소와 운영체제

- 사용자 : 컴퓨터를 사용하는 사람이나 장치, 다른 컴퓨터 등을 의미
- 소프트웨어 : 컴퓨터의 기능 수행에 필요한 모든 프로그램
- 하드웨어 : 기본 연산 자원을 제공하는 프로세서(CPU, 중앙처리장치), 메모리, 주변장치 등

운영체제의 개념

■ 컴퓨터 자원 관리 면에서 운영체제의 정의



- 조정자 : 운영 요소 사용을 제어하면서 사용자와 응용 프로그램 간에 통신할 수 있게 함.
작업을 할 수 있는 환경만 제공
- 자원 할당자 혹은 관리자 : 각 응용 프로그램에 필요한 자원 할당, 자원 할당 방법 결정
- 응용 프로그램과 입출력장치 제어자 : 다양한 입출력장치와 응용 프로그램 제어

운영체제의 개념

■ 운영체제의 정의와 역할

■ 정의

- 사용자와 하드웨어 사이의 중간 매개체로 응용 프로그램의 실행을 **제어**하고, 자원을 할당 및 **관리**하며, 입출력 제어 및 데이터 관리와 같은 **서비스**를 제공하는 소프트웨어

■ 역할

- 하드웨어 및 사용자, 응용 프로그램, 시스템 프로그램 사이에서 인터페이스를 제공
- 프로세서, 메모리, 입출력장치, 통신장치 등 컴퓨터 자원을 효과적으로 활용하려고 조정·관리
- 메일 전송, 파일 시스템 검사, 서버 작업 등 높은 수준의 서비스를 처리하는 응용 프로그램을 제어
- 다양한 사용자에게서 컴퓨터 시스템을 보호하려고 입출력을 제어하며 데이터를 관리

운영체제 발전의 목적

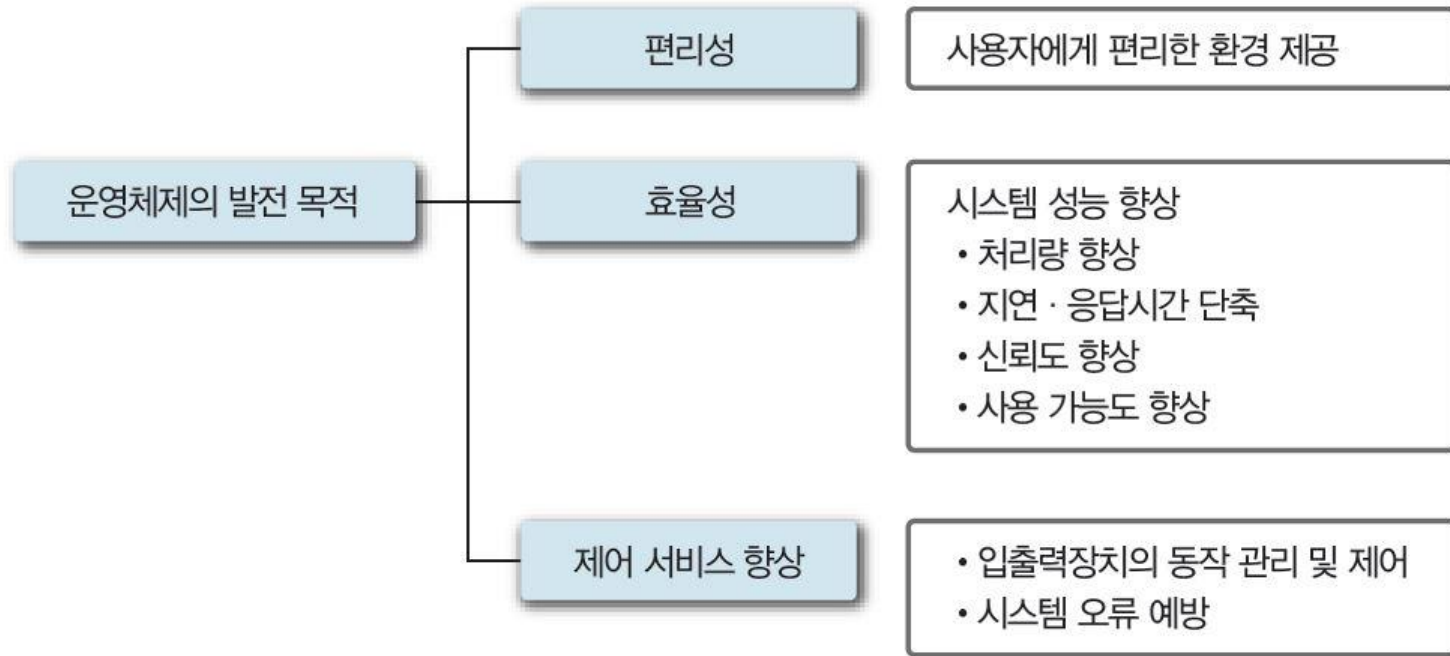


그림 2-3 운영체제의 발전 목적

■ 편리성 : 사용자에게 편리한 환경 제공

- 프로그램 개발 환경뿐만 아니라 응용 프로그램에 대한 사용자 인터페이스, 즉 사용자와 컴퓨터 시스템이 정보 및 명령을 상호 교환할 수 있는 인터페이스 제공

운영체제 발전의 목적

■ 효율성 : 시스템 성능 향상

- 사용자가 많은 대형 컴퓨터 시스템에서 특히 중요. 운영체제는 각 프로그램을 유기적으로 결합하여 시스템 전체 성능을 향상
- 시스템 성능의 평가 기준
 - 처리량
 - 지연·응답시간(턴 어라운드 타임)
 - 신뢰도
 - 사용 가능도(가동률)

■ 제어 서비스 향상

- 시스템 확장, 효율적 운영을 위해 새로운 기능의 효과적인 개발을 허용하는 방법으로 발전
- 입출력장치의 동작 관리 및 제어, 시스템 오류 예방 등으로 컴퓨터 자원을 여러 사용자에게 효율적으로 할당하고 관리할 수 있도록 제어 서비스를 발전

운영체제의 기능

■ 운영체제의 역할에 따른 기능



그림 2-4 운영체제의 기능

자원관리

■ 자원

- 컴퓨터 시스템의 메모리, 프로세스, 장치, 파일 등 구성 요소

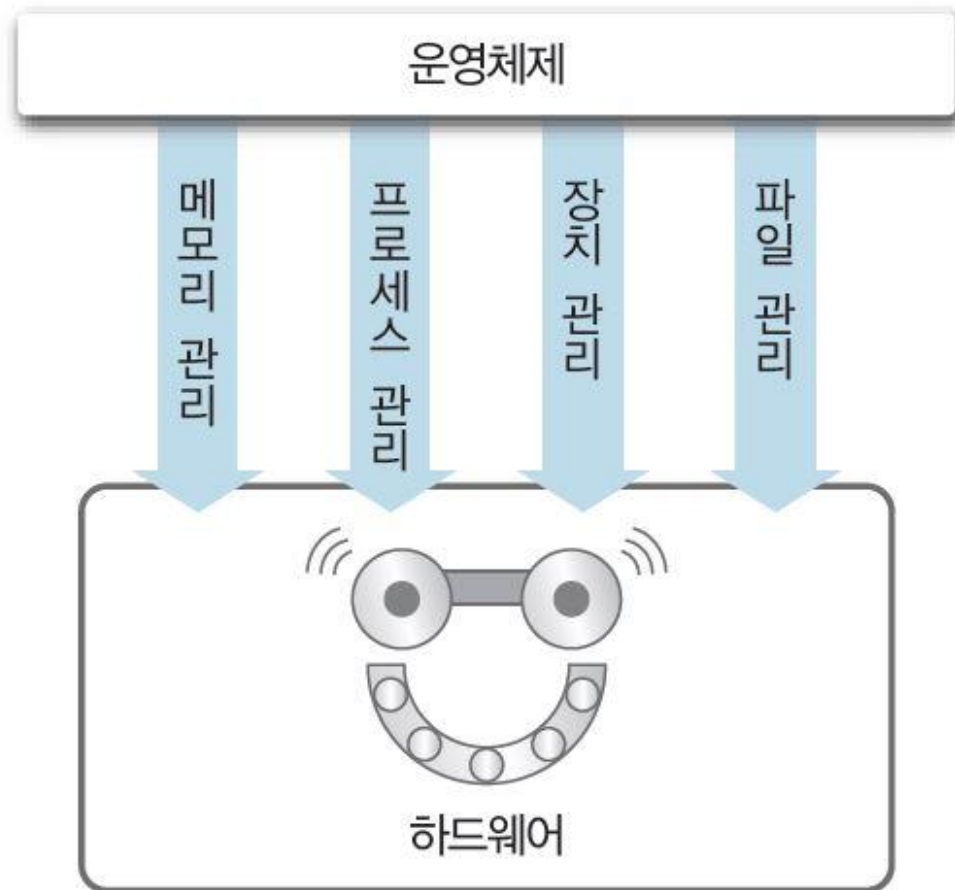


그림 2-5 운영체제의 자원 관리 기능

자원관리

■ 메모리 관리

- 메인 메모리 관리 : 프로세서가 직접 주소로 지정할 수 있는 유일한 메모리
- 메모리 관리의 기능
 - 메모리의 어느 부분을 사용하고, 누가 사용하는지 점검
 - 메모리에 저장할 프로세스 결정
 - 메모리를 할당하고 회수하는 방법 결정
- 보조기억장치 관리 : 메인 메모리는 공간이 제한되어 데이터와 프로그램을 계속 저장할 수 없어 보조기억장치 이용
- 보조기억장치 관리의 기능
 - 빈 여유 공간 관리
 - 새로운 파일 작성 시 저장 장소 할당
 - 메모리 접근 요청 스케줄링
 - 파일 생성하고 삭제

자원관리

■ 프로세스 관리

■ 프로세스

- 하나의 프로세스는 프로세서, 메모리, 파일, 입출력장치와 같은 자원으로 구성
- 자원은 프로세스 생성할 때 제공하거나 실행 중에도 할당 가능

■ 시스템

- 프로세스의 집합
- 시스템 코드 수행하는 운영체제 프로세스와 사용자 코드 수행하는 사용자 프로세스로 구분
- 모든 프로세스는 프로세서 분할 사용하여 병행 수행 가능

■ 프로세스 관리를 위한 운영체제의 기능

- 프로세스와 스레드 스케줄링
- 사용자 프로세스와 시스템 프로세스 생성, 제거
- 프로세스 중지, 재수행
- 프로세스 동기화 방법 제공
- 프로세스 통신 방법 제공
- 교착 상태^{deadlock}를 방지하는 방법 제공

자원관리

■ 주변장치(입출력장치) 관리

- 운영체제는 특수 프로그램인 장치 드라이브를 사용하여 입출력장치와 상호작용
- 장치 드라이버는 특정 하드웨어장치와 통신할 수 있는 인터페이스를 제공하므로 특정 하드웨어에 종속된 프로그램
- 주변장치(입출력장치) 관리를 위한 운영체제의 기능
 - 임시 저장^{buffer-caching} 시스템 기능 제공
 - 일반 장치용 드라이버 인터페이스 제공
 - 특정 장치 드라이버 제공

자원관리

■ 파일(데이터) 관리

- 입출력 파일의 위치, 저장, 검색 관리 의미
- 컴퓨터 시스템은 물리적으로 다양한 형태로 파일 저장 가능
- 운영체제는 데이터의 효율적 사용을 위해 단일화된 저장 형태 제공
- 운영체제는 파일의 용이한 사용을 위해 보통 디렉터리로 구성, 다수의 사용자가 여기에 접근하려고 할 때는 이 접근을 제어
- 파일 관리를 위한 운영체제의 기능
 - 파일 생성, 삭제
 - 디렉터리 생성, 삭제
 - 보조기억장치의 파일 맵핑
 - 안전한(비휘발성) 저장장치에 파일 저장

시스템 관리

■ 시스템 보호(사용자 권한 부여)

- 보호 : 컴퓨터 자원에서 프로그램, 프로세스, 사용자의 접근 제어 방법
- 운영체제는 파일 사용 권한 부여, 데이터 암호화 등 서비스를 제공, 데이터와 시스템 보안
- 컴퓨터 시스템에서는 여러 프로세스 동시 실행 가능하므로 상호 보호해야 함
- 네트워크로 파일 공유 사이트에 접속 시 다른 사용자의 프로그램에서 보호

■ 네트워킹(통신)

- 프로세서는 다양한 방법으로 구성된 네트워크 이용, 완전 접속과 부분 접속 방법으로 연결
- 연결된 프로세서가 통신을 할 때는 경로 설정, 접속 정책, 충돌, 보안 등 고려(운영체제가 관리)

■ 명령 해석기

- 명령 해석기 `command interpreter`는 운영체제에서 중요한 시스템 프로그램
- 대화형으로 입력한 명령어를 이해하고 실행하는 사용자와 운영체제의 인터페이스
- 사용자가 입력한 명령은 제어문으로 운영체제에 전달하는데, 이 전달을 명령 해석기가 담당
- 인터페이스 역할을 할 뿐 운영체제는 아님
- 커널과 분리하는 것이 좋음(명령 해석기의 인터페이스 변경 가능)
 - 분리하지 않으면 사용자가 커널의 코드를 변경할 수 없어 인터페이스를 변경 불가

운영체제의 발전 과정과 유형

■ 운영체제의 발전 과정

연도	운영체제	특징
1940년대	운영체제 없음(작업별 순차 처리)	<ul style="list-style-type: none">• 기계어를 직접 사용• 단순 순차(직렬) 처리
1950년대	일괄 처리 시스템	<ul style="list-style-type: none">• 운영체제의 효시인 IBM 701 개발• 작업별로 일괄 처리• 버퍼링, 스펀링 방법 등장
1960년대	<ul style="list-style-type: none">• 다중 프로그래밍 시스템• 시분할 시스템• 다중 처리 시스템• 실시간 처리 시스템	<ul style="list-style-type: none">• 가상 기억장치 등장• 다중 프로그래밍, 다중 처리, 시분할 처리 등 개념 등장• 운영체제를 고급 언어로 작성• 데이터 통신 지원용 운영체제 사용
1970년대 초반	<ul style="list-style-type: none">• 다중 모드 시스템• 범용 시스템	<ul style="list-style-type: none">• 일괄 처리, 시분할 처리, 실시간 처리, 다중 프로그래밍 등을 제공하는 다중 모드 시스템 등장• 장치의 독립성 제공• TCP/IP 통신 표준 활성화• 운영체제가 네트워크와 보안을 아우르는 수준으로 발전
1970년대 중반	분산 처리 시스템	<ul style="list-style-type: none">• 각종 응용 프로그램 개발 및 데이터베이스 활용 확대• 네트워크 기술의 발전• 하드웨어에 운영체제 개념이 포함된 펌웨어 개념 등장

운영체제의 발전 과정

■ 운영체제의 발전 과정

연도	운영체제	특징
1990년대	병렬 계산과 분산 계산	<ul style="list-style-type: none">• 월드와이드웹의 등장으로 분산 컴퓨팅 증가• GUI 강화• 개인용과 서버용 운영체제의 보편화
2000년대 이후	<ul style="list-style-type: none">• 모바일 및 임베디드• 가상화 및 클라우드 컴퓨팅	<ul style="list-style-type: none">• 네트워크 기반의 분산 및 병렬 운영체제의 보편화• 모바일 장치와 가전제품을 위한 모바일 및 임베디드 운영체제의 보편화• 다양한 기능, 확장성과 호환성 극대화• 다양한 통신망의 확대와 개발형 시스템 발달• 여러 운영체제가 한 시스템의 자원을 공유할 수 있게 해주는 서버 가상화 기술의 확산• 컴퓨팅 자원, 스토리지, 소프트웨어 등을 사용자에게 서비스 형태로 제공하는 클라우드 컴퓨팅의 등장

운영체제의 발전 과정

■ 1940년대 : 운영체제 없음(작업별 순차 처리)

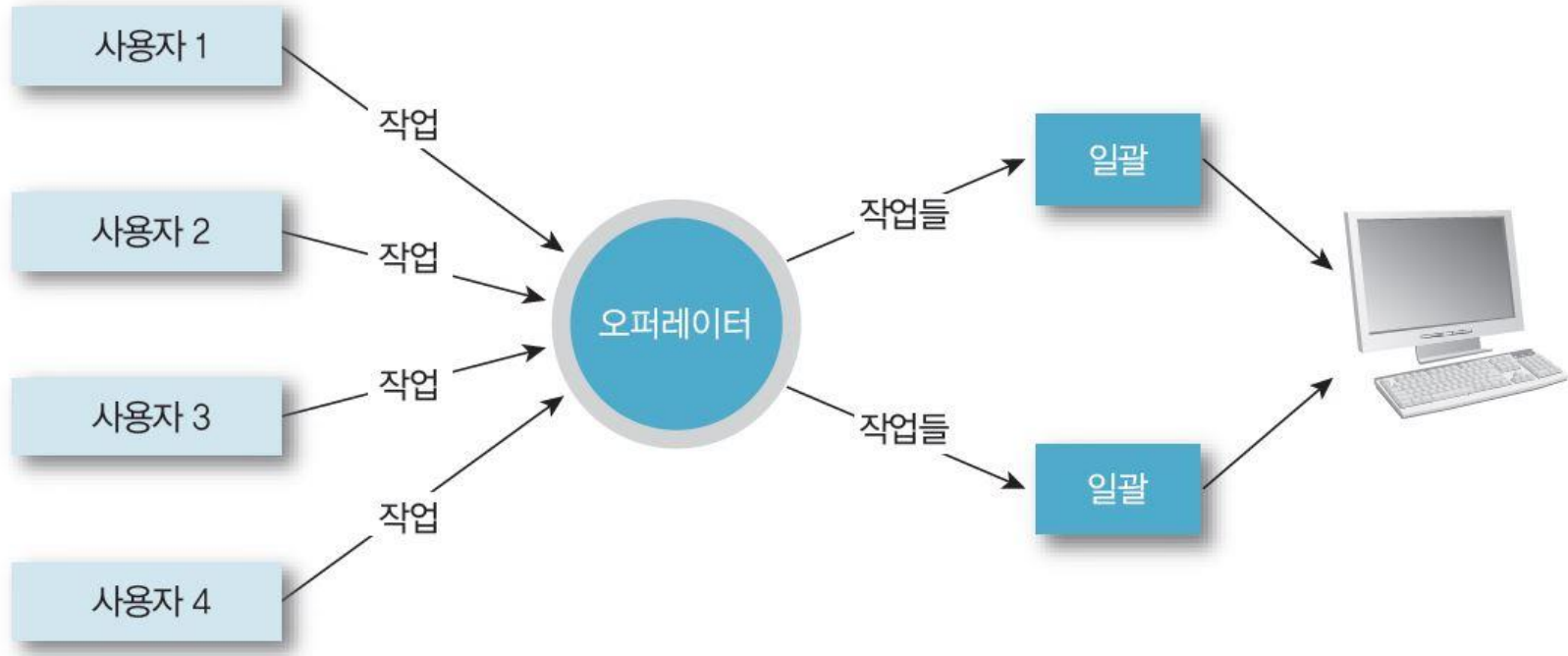
- 사용자가 기계어로 직접 프로그램 작성, 실행하는 작업별 순차 처리 시스템 사용
- 운영체제 개념 존재하지 않음
- 컴퓨터에 필요한 모든 작업 프로그램에 포함
- 카드 판독기에 판독의 시작·종료 시점, 데이터 해석 방법 등 포함
- 프로세서에는 명령어 저장 방법, 계산 대상, 결과 저장 위치와 방법, 출력 시점, 위치 등 모두 명령어로 명시적으로 표현
- 모든 작업을 예약으로 진행하여 문제가 발생

운영체제의 발전 과정

■ 1950년대 : 일괄 처리 시스템

- IBM 701 : 1952년 초 자동차 제조회사 GM에서 운영체제의 효시 개발
- IBM 704 : 1955년 GM과 북아메리카 항공사가 공동 개발
- IBM 704 자체 운영체제 : 1957년까지 IBM 사용자협회에서 개발
- 초기 운영체제인 일괄 처리 시스템 batch processing system
 - 작업을 올리는 시간과 해제하는 시간 줄이는 데 관심(일괄 처리, 버퍼링, 스푼링 등 방법 도입)
- 일괄 처리
 - 일괄 batch 처리는 직렬 처리 기술과 동일
 - 작업 준비 시간을 줄이려고 데이터가 발생할 때마다 즉시 처리하지 않고 데이터를 일정 기간 또는 일정량이 될 때까지 모아 두었다가 한꺼번에 처리
- 일괄 처리 장점
 - 많은 사용자와 프로그램이 컴퓨터 자원 공유
 - 컴퓨터 자원을 덜 사용 중일 때는 작업 처리 시간 교대 가능

운영체제의 발전 과정



■ 일괄 처리 단점

- 준비 작업들의 유형이 동일해야 하고, 작업에 모든 유형의 입력 불가능
- 입출력장치가 프로세서보다 속도 느려 프로세서의 유휴 상태 발생
- 작업 우선순위 부여 곤란.
- 문제점 보완 위해 모니터링, 버퍼링, 스펙링 등 여러 방법 등장

운영체제의 발전 과정

■ 버퍼링 buffering

- 유휴시간이 없도록 입/출력 장치 별로 입출력 버퍼 두어, 프로세서에서 연산 할 때 동시에 다른 작업 입/출력하는 아주 간단한 방법

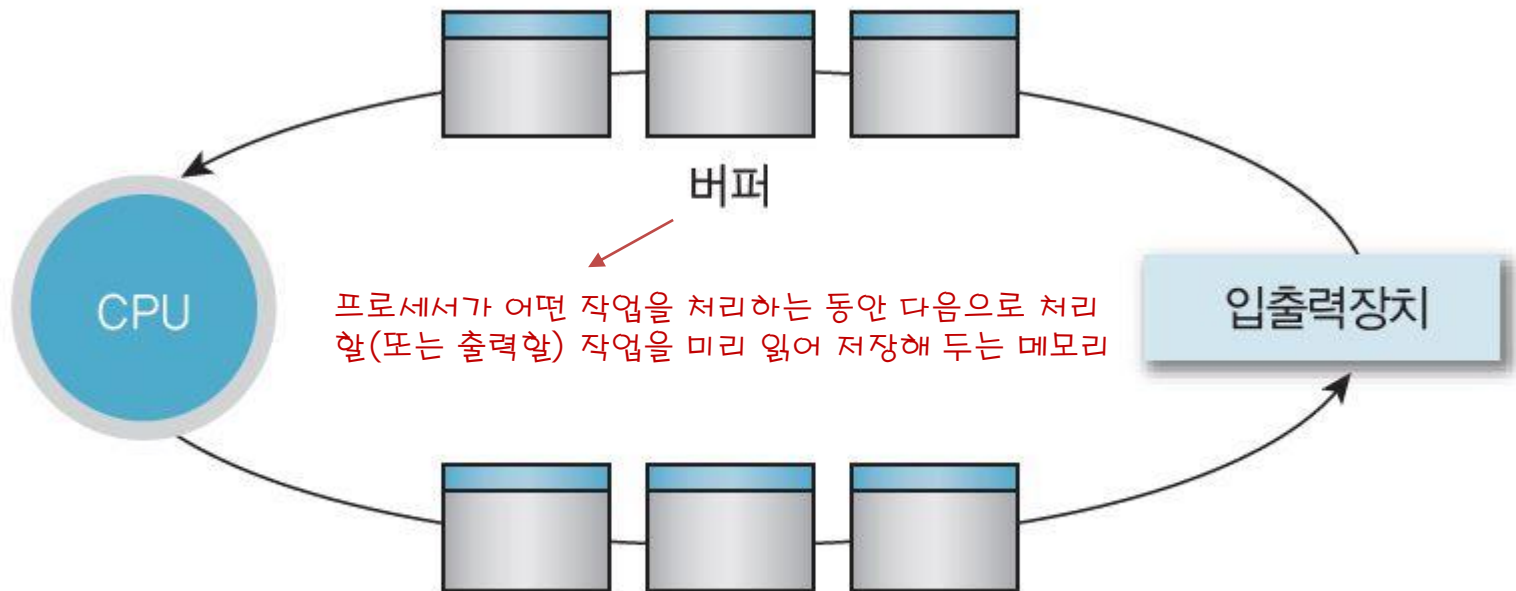


그림 2-7 버퍼링

운영체제의 발전 과정

- 스푼링 spooling, simultaneous peripheral operation on-line
 - 속도가 빠른 디스크를 버퍼처럼 사용 입출력장치에서 미리 읽는 것
 - 버퍼링이 컴퓨터 하드웨어의 일부인 버퍼를 사용 한다면, 스푼링은 별개의 오프라인 장치 사용
 - 버퍼링이 하나의 입출력 작업과 그 작업의 계산만 함께 할 수 있는 반면, 스푼링은 여러 작업의 입출력과 계산을 함께 할 수 있음
 - 프로세서에 일정한 디스크 공간, 테이블만 있으면 하나의 계산 작업과 다른 입출력 작업 중복 처리
 - 프로세서와 입출력장치가 고효율로 작업하게 함
 - 성능에 직접적으로 도움

운영체제의 발전 과정

■ 1960년대 : 다중 프로그래밍·시분할·다중 처리·실시간 시스템

- 장치 독립성을 이용한 편리한 하드웨어 관리와 다중 프로그래밍
 - 여러 프로그램을 메모리에 나눠 적재한 후 프로세서를 번갈아 할당 프로세서 사용 극대화하여
 - 여러 프로그램을 동시에 실행
 - 장치 독립성 : 프로그램을 다른 입출력장치와 함께 실행할 수 있는 것
- 시분할, 다중 처리, 실시간을 이용한 시스템의 처리 능력 향상
 - 시분할 시스템 : 다중 프로그래밍 시스템에 프로세서 스케줄링이라는 개념을 더한 것
 - 다중 처리 시스템 : 하나의 시스템에서 프로세서를 여러 개 사용하여 처리 능력을 높인 것
 - 실시간 처리 시스템 : 즉시 응답

운영체제의 발전 과정

- **1970년대 중반~1990년대 : 분산 처리 시스템, 병렬 계산과 분산 계산**
 - 컴퓨터 네트워크와 온라인 처리 방법 널리 사용
 - 네트워크 이용하여 멀리 떨어진 컴퓨터 사용
 - 마이크로프로세서가 등장, 개인용 컴퓨터 보유
 - 사용자가 지역적으로 원격의 여러 시스템과 통신할 수 있어 정보 보호가 주요 관심사
 - 1970년대 : 명령어 중심의 시스템 사용
 - 1980년대 : 사용자에게 편리 한 메뉴 지향적인 시스템
 - 1990년대 : GUI(Graphical User Interface) 시스템
 - 분산 처리 개념을 확립하여 데이터 발생하는 곳으로 컴퓨터의 능력을 가져오는 데 관심

운영체제의 발전 과정

■ 2000년대 이후 : 모바일 및 임베디드, 가상화 및 클라우드 컴퓨팅

- 21세기에 접어들어 스마트폰이나 태블릿 같은 모바일 기기 대중화
- 모바일 운영체제^{Mobile Operating System} : 모바일장치나 정보 기기 제어 운영체제
 - 스마트폰용 : 노키아(심비안), 구글(안드로이드^{android}), 애플(iOS), RIM(블랙베리 OS), 마이크로소프트의 윈도우 등이 대표적이다.
- 사물 인터넷^{IoT, Internet of Things} 기술 등장 : 각종 사물에 컴퓨터칩과 통신 기능 내장 인터넷에 연결
 - 초기에는 시스코 등 네트워크 기업, 나중에는 인텔, 퀄컴을 비롯한 반도체 칩 판매사가 사물인터넷을 전파하다가 이제는 애플이나 삼성전자 같은 제조사가 사물인터넷과 관련된 제품의 청사진과 플랫폼 구상 중
 - 특히 구글이나 아마존 등은 인터넷과 클라우드 기반 플랫폼에서 우위를 바탕으로 사물 인터넷 시장 선도하려고 하루가 멀다 하고 다양한 관련 기술과 서비스 발표.

운영체제의 유형



그림 2-11 운영체제의 유형

운영체제의 유형

■ 다중 프로그래밍 시스템

- 프로세스가 다른 작업 수행 시 입출력 작업 불가능하여 프로세서와 메인 메모리의 활용도 떨어지는 일괄 처리 시스템의 큰 문제를 다중 프로그래밍 도입하여 해결
- 프로세서가 유휴 상태일 때 실행 중인 둘 이상의 작업이 프로세서를 전환 (인터리빙)하여 사용할 수 있도록 동작

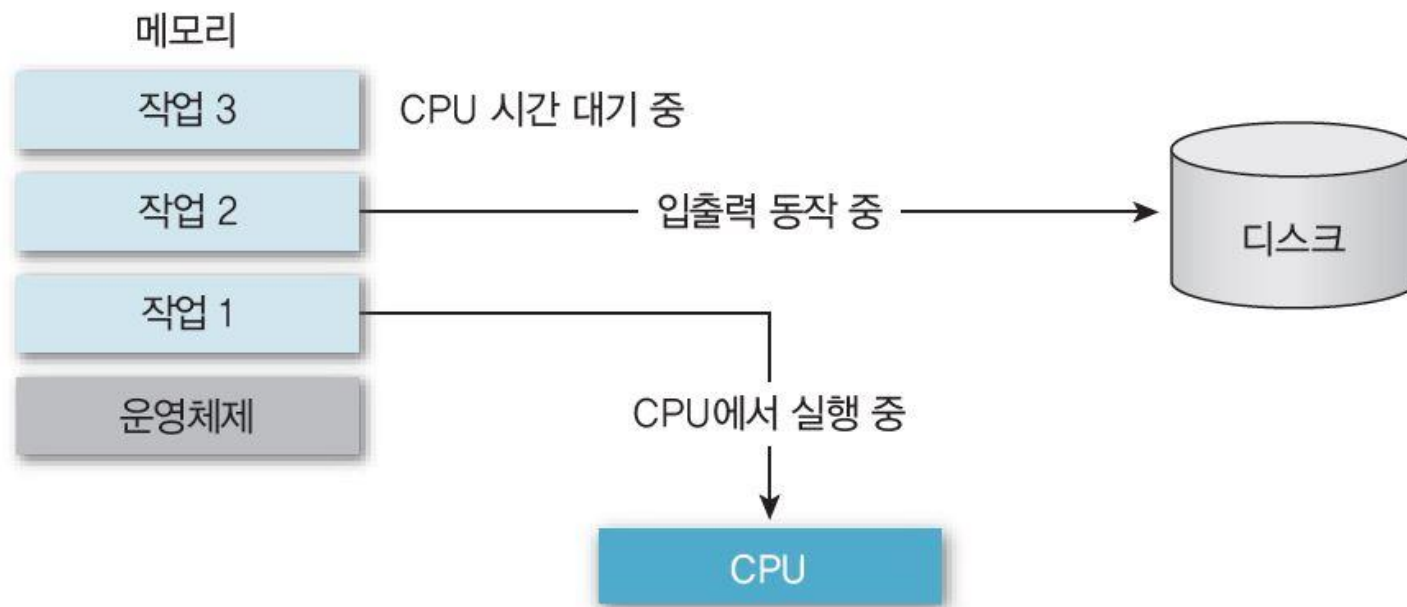


그림 2-12 다중 프로그래밍 시스템

운영체제의 유형

■ 다중 프로그래밍의 특징

- 높고 효율적인 프로세서 사용률(효율적인 운영) 증가
- 많은 사용자의 프로그램이 거의 동시에 프로세서를 할당받는 듯한 느낌
- 다중 프로그래밍 운영체제는 아주 복잡
- 여러 작업을 준비 상태로 두려면 이를 메모리에 보관, 일정 형태의 메모리를 관리해야 함
- 여러 작업이 수행할 준비를 갖추고 있으면, 이 중 하나를 선택하는 결정 방법 필요

(인터럽트 이용 수행하는 프로세서 스케줄링의 다중 프로그래밍으로, 현재 운영체제의 중심 주제)



그림 2-13 다중 프로그래밍 시스템의 처리 방법 예

운영체제의 유형

■ 시분할 시스템 TSS, Time Sharing System

- 다중 프로그래밍을 논리적으로 확장한 개념, 프로세서가 다중 작업을 교대로 수행
- 다수의 사용자가 동시에 컴퓨터의 자원을 공유할 수 있는 기술
- CTSS Compatible Time Sharing System : MIT 에서 개발, 1961년 IBM 709에 탑재, 사용
- 1970년 초까지는 시분할 시스템 만들기가 아주 어렵고 비용도 많이 들어 일반화 못함
- 각 프로그램에 일정한 프로세서 사용 시간 또는 규정 시간량 할당, 컴퓨터와 대 화하는 형 식으로 실행
- 여러 사용자에게 짧은 간격으로 프로세서 번갈아 할당, 마치 자기 혼자 프로세서를 독점하 고 있는 양 착각하게 하여 여러 사용자가 단일 컴퓨터 시스템을 동시 사용 가능

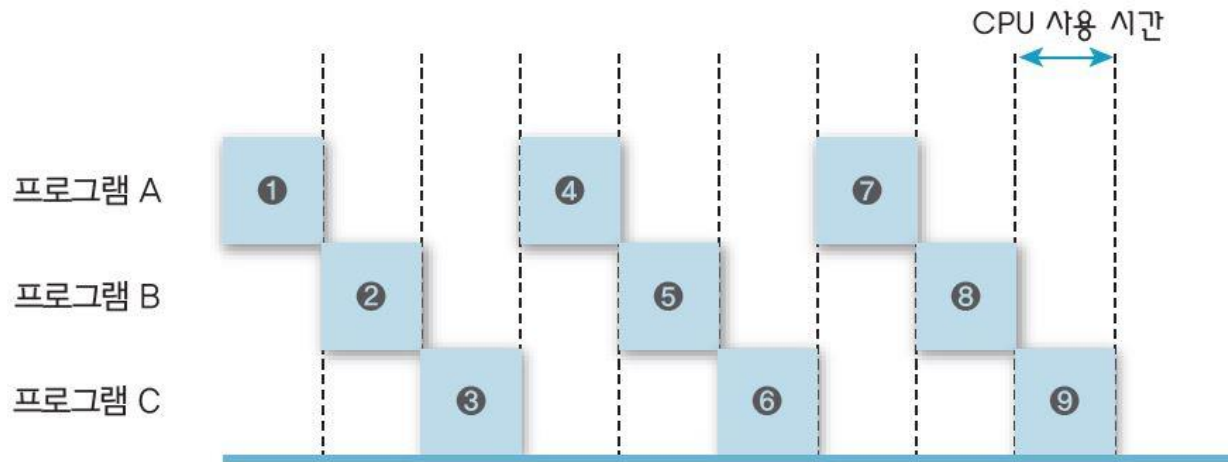


그림 2-14 시분할 시스템의 처리 방법 예

운영체제의 유형

- 다중 프로그래밍 시스템과 시분할 시스템 특징
 - 메모리에 여러 프로그램을 적재하므로 메모리 관리 필요
 - 어떤 프로그램을 먼저 실행할지 결정하는 스케줄링 개념 필요
 - 다중 프로그래밍 시스템의 목표 : 프로세서 사용 최대화
 - 시분할 시스템의 목표 : 응답시간 최소화

표 2-2 시분할 시스템의 장점과 단점

장점	<ul style="list-style-type: none">• 빠른 응답 제공• 소프트웨어의 중복 회피 가능• 프로세서 유휴시간 감소
단점	<ul style="list-style-type: none">• 신뢰성 문제• 보안 의문 및 사용자 프로그램과 데이터의 무결성• 데이터 통신의 문제

운영체제의 유형

■ 다중 처리 multiprocessing 시스템

- 단일 컴퓨터 시스템 내에서 둘 이상의 프로세서 사용, 동시에 둘 이상의 프로세스 지원
- 여러 프로세서와 시스템 버스, 클록, 메모리와 주변장치 등 공유
- 빠르고, 프로세서 하나가 고장 나도 다른 프로세서 사용하여 작업 계속, 신뢰성 높음
- 프로세서 간의 연결, 상호작용, 역할 분담 등을 고려해야 함
- 다중 처리 시스템을 구성하는 방법에는 비대칭(주종)적 구성과 대칭적 구성이 있음

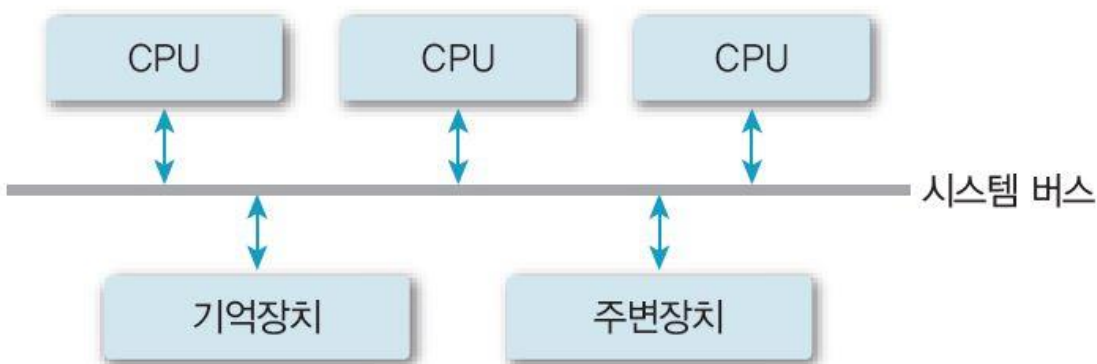


그림 2-15 다중 처리 시스템

운영체제의 유형

■ 실시간 처리 시스템 real time processing system

- 데이터 처리 시스템으로 정의
- 입력에 응답하는 데 필요한 시간 간격이 너무 짧아 환경 제어
- 온라인 시스템은 실시간으로 할 필요 없지만, 실시간 처리 시스템은 항상 온라인 상태
- 입력 및 업데이트된 정보 요구 처리 후 디스플레이에 응답하는 시스템에 소요 시간을 반응(응답) 시간으로 함
- 반응시간은 프로세서에 이미 고정(반응시간이 온라인 처리에 비해 매우 짧음)
- 더 높은 적시 응답을 요구하는 장소에서 사용하거나 데이터 흐름 또는 프로세서 연산에 엄격한 시간 요구가 있을 때 사용 가능, 전용 응용 프로그램의 제어장치로도 사용
- 고정 시간 제약을 잘 정의하지 않으면 시스템 실패.
- 실시간 처리 시스템의 두 가지 유형
 - 경성 실시간 처리 시스템 hard real time processing system
 - 작업의 실행 시작이나 완료에 대한 시간 제약 조건을 지키지 못할 때 시스템에 치명적인 영향을 주는 시스템
 - 무기 제어, 발전소 제어, 철도 자동 제어, 미사일 자동 조준 등이 이에 해당
 - 보장되는 컴퓨팅, 시간의 정확성과 컴퓨팅 예측성을 갖게 해야 함
 - 연성 실시간 처리 시스템 soft real time processing system
 - 작업 실행에서 시간 제약 조건은 있으나, 이를 지키지 못해도 전체 시스템에 치명적인 영향을 미치지 않는 시스템
 - 동영상은 초당 일정 프레임 frame 이상의 영상을 재생해야 한다는 제약이 있으나, 일부 프레임을 건너뛰어도 동영상을 재생 시스템에는 큰 영향을 미치지 않음

운영체제의 유형

■ 분산 처리 시스템 distributed processing system

- 시스템마다 독립적인 운영체제와 메모리로 운영, 필요 시 통신하는 시스템
- 사용자에게는 중앙집중식 시스템처럼 보이는데, 다수의 독립된 프로세서에서 실행
- 데이터를 여러 위치에서 처리·저장, 여러 사용자가 공유
- 하나의 프로그램을 여러 프로세서에서 동시에 실행

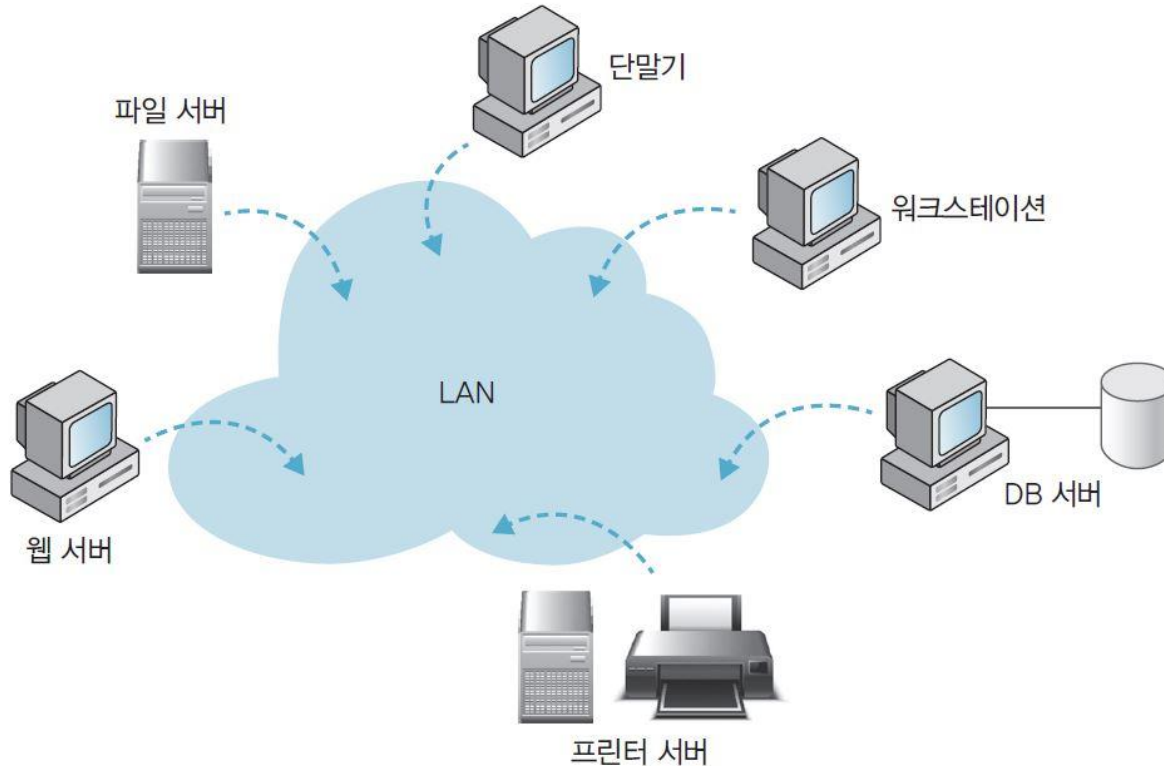


그림 2-16 분산 처리 시스템

운영체제의 서비스

■ 운영체제의 서비스 제공

- 부팅 서비스 : 컴퓨터 하드웨어 관리, 프로그램을 실행할 수 있도록 컴퓨터에 시동
- 사용자 서비스 : 프로그래머가 프로그래밍 작업을 쉽게 수행할 수 있도록 함
- 시스템 서비스 : 시스템의 효율적인 동작 보장
- 시스템 호출 : 프로그램이 운영체제의 기능을 서비스 받을 수 있는 프로그램과 운영체제 간의 인터페이스 제공

부팅 서비스

■ 부팅^{booting} 또는 부트스트래핑^{bootstrapping}

- 운영체제를 메인 메모리에 적재하는 과정
- 부트 로더는 부트스트랩 로더^{bootstrap loader} 줄인 말로 하드디스크와 같은 보조기억장치에 저장된 운영체제를 메인 메모리에 적재하는 ROM에 고정시킨 소규모 프로그램

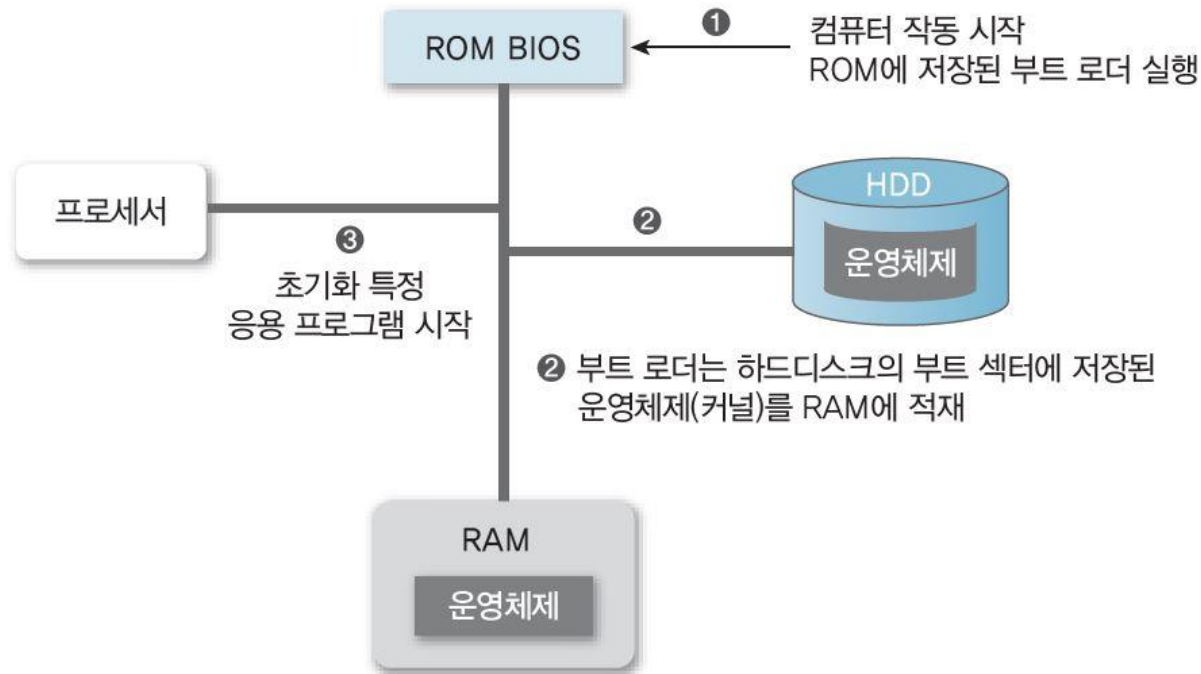


그림 2-17 부팅 과정

사용자 서비스

■ 사용자 인터페이스 제공

- 사용자 인터페이스 : 사용자와 컴퓨터 간의 상호작용 발생 공간(CLI, 메뉴, GUI 등 구현)

- CLICommand Line Interface(명령 라인 인터페이스)

사용자가 키보드 등으로 명령어 입력하여 시스템에서 응답 받은 후, 또 다른 명령어를 입력하여 시스템을 동작하게 하는 텍스트 전용 인터페이스. 사용자가 프롬프트에서 명령어를 입력하여 컴퓨터와 상호작용할 수 있고, 명령어 입력한 후 반드시 **Enter**를 눌러야 함

- 메뉴 인터페이스

메뉴 등을 사용하여 시스템과 상호작용. 사용 매우 편리, 배우거나 기억해야 할 명령 없음
iPad나 휴대폰, 현금 자동 인출기ATM 등이 대표적

- GUIGraphical User Interface(그래픽 사용자 인터페이스)

윈도우 환경에서 사용자에게 정보와 작업을 표현하는 텍스트, 레이블이나 텍스트 탐색과 함께 그래픽 아이콘과 시각적 표시기, 버튼이나 스크롤바와 같은 위젯^{widget} 그래픽 제어 요소를 사용 컴퓨터와 상호작용 할 수 있는 가장 보편적인 유형
마이크로소프트의 윈도우나 애플의 맥 OS에 사용하는 방법이 대표적

사용자 서비스

■ 프로그램 실행

- 프로그램 실행하려면 먼저 메모리에 적재, 프로세서 시간 할당
- 운영체제는 프로그램을 실행하려고 메모리 할당이나 해제, 스케줄링등 중요 작업 처리

■ 입출력 동작 수행

- 수행 중인 프로그램은 입력이 필요, 사용자가 제공하는 입력 처리 후에는 출력을 생성
- 운영체제는 입출력 동작 직접 수행할 수 없는 사용자 프로그램의 입출력 동작 방법 제공

■ 파일 시스템 조작

- 사용자는 디스크에서 파일 열고, 저장, 삭제하는 등 다양하게 파일 조작
- 디스크에 파일을 저장하면 특정 블록에 할당 저장, 파일을 삭제하면 파일 이름 제거되면서 할당한 블록이 자유롭게 됨.
- 운영체제는 파일 시스템 조작 서비스 제공, 사용자가 파일 관련 작업을 쉽게 할 수 있게 함.

사용자 서비스

■ 통신(네트워크)

- 프로세스가 다른 프로세스와 정보를 교환하는 방법
 - 동일한 컴퓨터에서 수행하는 프로세스 간의 정보 교환
 - 두 번째는 네트워크로 연결된 컴퓨터 시스템에서 수행하는 프로세스 간의 정보 교환
- 운영체제는 다중 작업 환경에서 공유 메모리를 이용하거나 메시지 전달로 다양한 유형의 프로세스와 통신 지원

■ 오류 탐지

- 운영체제는 가능한 모든 하드웨어와 소프트웨어 수준에서 오류 탐지, 시스템 모니터링하여 조정함으로써 하드웨어 문제 예방
- 입출력 장치에 관련된 오류와 메모리 오버 플로, 하드디스크의 불량 섹터 검출, 부적당한 메모리 접근과 데이터 손상 등
- 운영체제는 다음 오류 유형을 감지한 후 유형별로 적절히 조치
 - 프로세서, 메모리 하드웨어와 관련된 오류 : 기억장치 메모리 오류, 정전
 - 입출력장치 오류 : 테이프의 패리티 오류, 카드 판독기의 카드 체증^{jam}, 프린터의 종이 부족
 - 사용자 프로그램 오류 : 연산의 오버플로, 부적당한 기억장치 장소 접근, 프로세서 시간 과다 사용

시스템 서비스

- **시스템 서비스 : 사용자가 아닌 시스템 자체의 효율적 동작 보장하는 기능**
 - 여러 사용자가 사용하는 시스템에서 컴퓨터 자원 공유하여 시스템 자체의 효율성 높임
- **자원 할당**
 - 운영체제는 다수의 사용자나 작업 동시 실행 시 운영체제가 자원을 각각 할당하도록 관리
 - 프로세서 사이클, 메인 메모리, 파일 저장 장치 등은 특수한 할당 코드를 갖지만, 입출력장치 등은 더 일반적인 요청과 해제 코드 가질 수 있음
- **계정**
 - 운영체제는 각 사용자가 어떤 컴퓨터 자원을 얼마나 많이 사용하는지 정보 저장 추적
 - 이 정보는 사용자 서비스 개선을 위해 시스템 재구성하는 연구자에게 귀중한 도구가 됨
- **보호와 보안**
 - 운영체제는 다중 사용자 컴퓨터 시스템에 저장된 정보 소유자의 사용을 제한
 - 서로 관련이 없는 여러 작업을 동시에 수행할 때는 한 작업이 다른 작업이나 운영체제를 방해하지 못하게 해야 함.
 - 보호 : 시스템 호출 하려고 전달한 모든 매개변수의 타당성 검사하고, 시스템 자원에 모든 사용자 접근을 제어하도록 보장하는 것
 - 보안 : 잘못된 접근 시도에서 외부 입출력장치 방어, 외부에 사용자 인증을 요구하는 것

시스템 호출system call

- 실행 중인 프로그램과 운영체제 간의 인터페이스, APIApplication Programming Interfaces라고도 함
- 사용자 프로그램은 시스템 호출을 하여 운영체제의 기능 제공 받음.
- 핵심 커널 서비스와 통신, 새로운 프로세스의 생성과 실행, 하드웨어 관련 서비스 등이 있음
- 시스템과 상호작용하는 동작은 대개 사용자 수준 프로세스에서는 사용할 수 없으나, 시스템 호출을 하여 운영체제에 서비스를 요청할 수 있음
- 시스템 호출 방법
 - 프로그램에서 명령이나 서브루틴의 호출 형태로 호출
 - 시스템에서 명령 해석기를 사용하여 대화 형태로 호출
- 운영체제가 제공하는 일반적인 시스템 호출
 - 프로세스 제어, 파일 조작, 장치 관리, 정보 유지 등

시스템 호출system call

호출 서비스	설명	
프로세스 제어	<ul style="list-style-type: none">• 종료와 취소• 적재^{load}와 실행• 프로세스 생성과 종료	<ul style="list-style-type: none">• 프로세스 속성 획득과 지정• 대기와 대기 이벤트, 신호 이벤트• 메모리 할당과 해제
파일 조작	<ul style="list-style-type: none">• 파일 생성과 삭제• 파일 열기^{open}와 닫기^{close}	<ul style="list-style-type: none">• 파일 읽기와 쓰기, 파일 재배치^{reposition}• 파일 속성 획득과 지정
장치 조작	<ul style="list-style-type: none">• 장치 요구와 해제• 장치 읽기와 쓰기, 재배치	<ul style="list-style-type: none">• 장치 속성 획득과 설정• 논리적 부착이나 장치 제거
정보 관리	<ul style="list-style-type: none">• 시간과 날짜의 설정과 획득• 데이터의 설정과 획득• 프로세스, 파일, 장치 속성의 설정과 획득	
통신	<ul style="list-style-type: none">• 통신 연결의 생성과 제거• 메시지의 송수신	<ul style="list-style-type: none">• 정보 상태 전달• 원격 장치의 부착 및 제거

최근 운영체제의 특징

■ 최근 운영 체제의 특징

■ 멀티미디어 자원의 관리

: 기존의 텍스트 위주의 자원에서 동영상, 사운드, 애니메이션 등의 여러 형태의 미디어 파일을 처리함

■ Windows 및 Graphics User Interface(GUI)의 일반화

: Windows 메뉴와 마우스 클릭으로 컴퓨터와 인간의 상호 작용이 가능함

■ 보안 기술 강화

: 인터넷 등의 통신망의 발달로 인한 접근제어 기술과 보안기술이 강화됨

■ 객체지향 기술 이용

: GUI, OLE에서 이용된 개념으로 모든 종류의 정보 객체 - 아이콘, 바탕화면, 문서, 동영상, 소리, 3D, 계속 갱신되는 뉴스, 컨트롤 등 대략 600개 이상의 종류-를 관리, 실행하기 위한 기술임

운영 체제의 개념

■ 최근 운영 체제의 특징

■ UNIX의 기능 확장 및 다중화

: UNIX의 기능이 다양하게 확장되었고 이러한 기능들은 Windows, Mac OS 등에서 이용되고 있으며, UNIX 자체도 다양한 버전이 출시됨

■ Windows의 대중화

: MICROSOFT사의 OS인 Windows제품군이 대중화되어 PC들 사이에 호환성에서 발전을 이룸

■ Linux의 확대

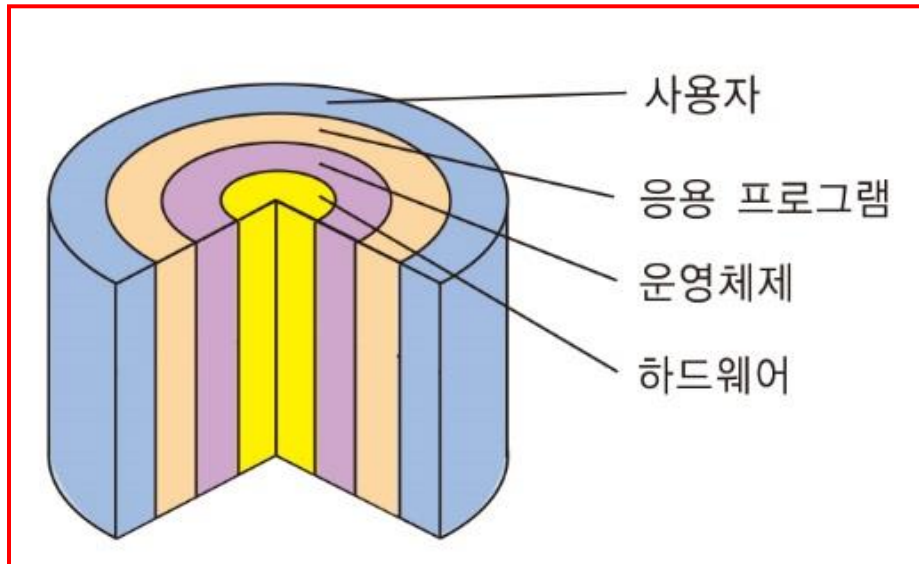
: PC용 UNIX라 할 수 있는 Linux가 open source 개념으로 급속하게 확산됨

운영체제 종류

운영 체제의 개념

■ 운영 체제(Operating System : OS)

- **OS**란 시스템의 자원인 CPU, RAM, HDD, I/O 장치, 네트워크 등을 효율적으로 관리하고 운영함으로써 사용자와 컴퓨터 하드웨어간의 인터페이스 역할을 함



운영체제의 개념도

운영 체제의 개념

■ OS의 목표와 기능

■ OS의 목표

- 컴퓨터 내의 하드웨어/소프트웨어 자원을 관리함
- 컴퓨터에 사용자가 쉽게 접근할 수 있는 인터페이스를 제공함
- 수행 중인 프로그램들의 효율적인 운영을 도움
- 작업 처리 과정중의 데이터를 공유함
- 입출력에 보조적인 기능을 수행함
- 오류가 발행하면 오류를 처리함

운영 체제의 개념

■ OS의 자원 관리

■ 프로세스(PROCESS)관리

- 프로세스란 현재 컴퓨터의 기억장치에서 수행 중인 프로그램의 상태를 말함
- OS는 이런 프로세스의 생성, 삭제, 동기화 등에 관여함

■ 주기억 장치 관리

: 주기억 장치의 할당과 회수를 관리함

■ 보조기억 장치 관리

: 보조기억 장치의 사용을 관리함

■ 입출력 장치 관리

: 입출력 장치를 관리함

■ 파일 관리

: 기억 장소의 할당, 빈 공간의 관리, 디스크의 스케줄링 등을 담당함

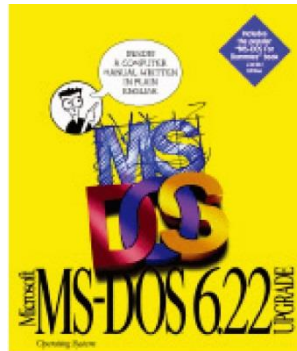
■ DOS(Disk Operating System)

- 대표적인 단일 사용자, 단일 태스크의 운영체제임
- MS의 MS-DOS, IBM의 PC-DOS 등이 대표적임
- 지금은 거의 사용하지 않지만 현재 쓰고 있는 Windows 시리즈에 많은 영향을 줌

■ DOS의 특징

- 초기의 IBM PC를 위해 만들어진 운영체제 임
- 텍스트 기반의 명령어를 직접 입력하는 방식으로 작동됨
- 저 용량 PC를 기준으로 만들어졌기 때문에 속도가 다른 운영체제에 비해 빠른 장점을 가짐
- 하지만 사용자가 명령어를 일일이 암기해야 하는 단점이 있음

DOS



MS-DOS 6.22 최종 버전

```
Data error reading drive A
Abort, Retry, Ignore, Fail?a
Program too big to fit in memory
A:\>
A:\>dir

Volume in drive A has no label
Volume Serial Number is AEAD-3267
Directory of A:\

SETRAMD  BAT           388   03-11-04   7:37p
CONFIG   SYS           139   03-11-04   9:38p
FINDRAMD EXE        8,855   04-23-98  10:22p
HIMEM    SYS        33,191   04-23-98  10:22p
RAMDRIVE SYS       12,863   04-23-98  10:22p
AUTOEXEC BAT         381   03-11-04   9:49p
DSCK     TA         772,688   05-01-13   3:03p
HDDREG   EXE       114,688   05-01-13   3:02p
HDDREG   LOG          88   05-01-13   3:16p
9 file(s)          948,913 bytes
0 dir(s)          129,024 bytes free

A:\>_
```

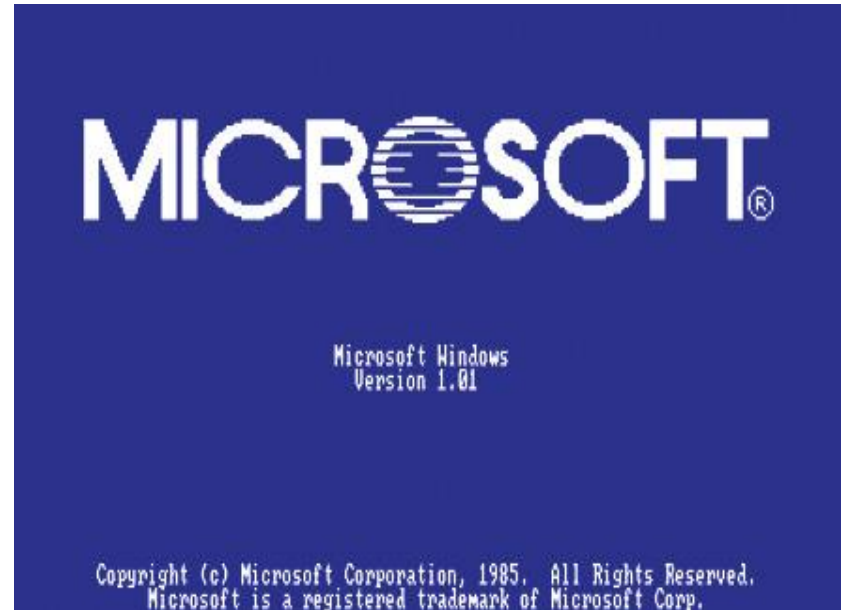
DOS의 디렉토리 화면

Windows XP

- **Windows** : 마이크로소프트사에서 만든 그래픽 사용자 인터페이스 (GUI) 운영 체제의 이름



Windows XP의 바탕 화면



Windows 1.0의 시작 화면

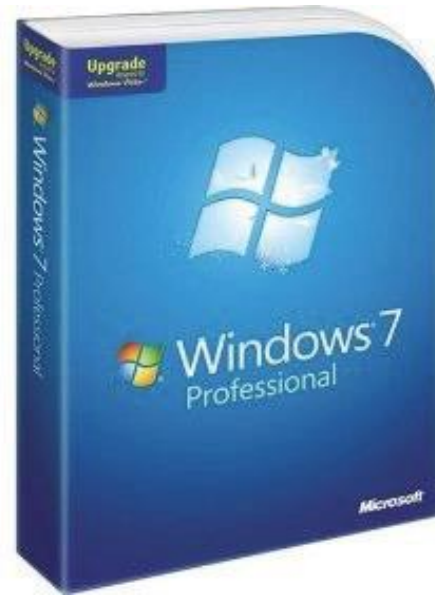
■ Windows의 역사

- 1985년 : 처음 발표됨
- 1995년: Windows 95를 발표하면서 DOS와 분리됨
- 1998년: 인터넷을 강화 시킨 98버전을 출시함
- 2001년 10월 : 개발 명 휘슬러인 Windows XP를 출시함
- 2009년 10월 : Windows 7 발표
- 2012년 10월 : Window 8 발표
- 2014년 10월 : Window 10 시험판 발표

Windows XP



Window XP



Window 7

Windows XP

■ Windows XP

New Experience에서 XP를 따옴

Windows ME계열과 Windows 2000을 통합하여 만들어짐

■ XP의 주요 특징

- 설계시의 중요한 부분은 기업용이나 전문가용으로 쓰이던 2000을 기반으로 설계되었기 때문에 뛰어난 안정성을 보장함
- 기존의 Windows시리즈보다 화려한 인터페이스를 제공함
- 기본적으로 제공되는 프로그램들이 개선되고, (예 > CD-R(W)나 5.1채널의 사운드, 채팅, 원격제어 관리 등을 지원함) 사용자 인터페이스가 화려해진 만큼 보다 높은 사양의 컴퓨터를 필요로 함
- 인터넷을 기반으로 디자인 되어, 구입 후 지원, 업그레이드가 인터넷을 기본으로 설정됨

Windows XP



Windows XP의 로그인 화면

Windows 7

■ Windows 7

- Windows 7은 빠르고 안정적인 성능과 함께 사용자가 보다 편리한 방법으로 PC를 사용하여 작업할 수 있도록 설계됨



Windows 7의 바탕 화면

■ Windows 7의 시스템 요구사항

- 1GHz 이상의 32비트(x86) 또는 64비트(x64) 프로세서
- 1GB RAM(32비트) 또는 2GB RAM(64비트)
- 하드디스크 16GB(32비트) 또는 20GB(64비트)
- WDDM 1.0 이상의 드라이버가 있는 DirectX 9 그래픽장치



Windows 7의 로그인 화면

■ Windows 7 주요 특징

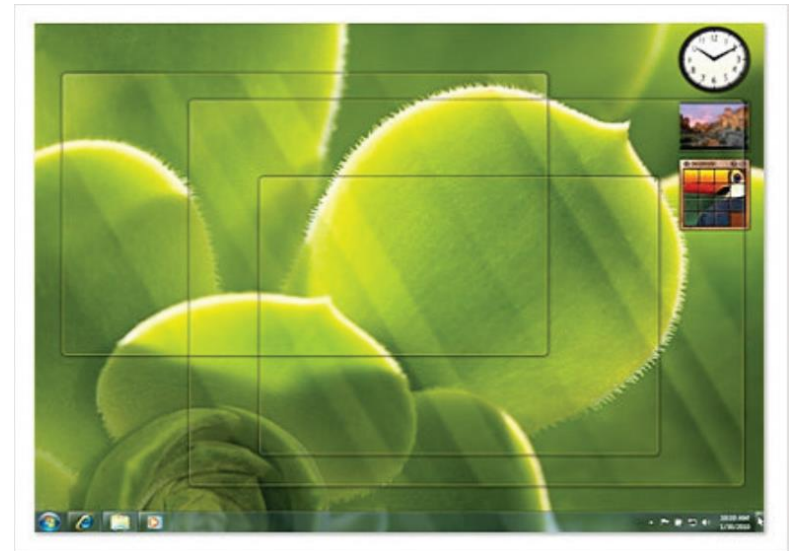
- 속도가 빠르고 최소 설치 요구사항이 가벼워 짐
- 네트워킹 기능이 단순화되고 향상됨
- BitLocker 기본탑재로 보안기능 강화됨
- 가상화 기능으로 호환성 강화됨
- 사용자 인터페이스 편리, 장애 우들을 위한 기능 향상됨
- 미디어 센터의 타임머신 기능을 탑재하여 멀티미디어 기능이 크게 향상 됨

Windows 7

- **Windows 7 주요 기능**
 - 홈 그룹 기능



- 바탕화면 기능



Windows 7

- **Windows 7 주요 기능**
 - 멀티미디어 재생 기능



- 터치기능



Windows 8

■ Windows 8

- Windows 8은 2012년 10월에 마이크로소프트의 새로운 운영 체제로 발표됨. 윈도우폰에 쓰인 메트로 사용자 환경이 PC에 적용됨



Windows 8

Windows 8과 Window 10

■ Windows 8의 특징

- PC, 노트북 외에 태블릿이나 스마트폰과 같은 모바일 기기에서도 사용할 수 있는 MS의 첫 번째 통합 운영체제임
- 컴퓨터의 부팅시간(약 8초)과 종료 시간이 종전의 절반으로 짧아짐
- 첫 화면이 타일 모양의 아이콘으로 구성되어 있어 한 번의 클릭으로 원하는 어플리케이션과 콘텐츠에 바로 접속할 수 있음

Windows 8과 Window 10

■ Windows 8의 특징 (계속)

- Window 7까지 필요했던 가상디스크 구동 프로그램이 기본으로 내장되어 있어 별도의 설치가 불필요함
- 터치에 최적화된 새로운 인터넷 익스플로러 10을 탑재했으며, 클라우드 서비스인 SkyDrive도 기본으로 제공함

Windows 8과 Window 10

■ Windows 10

- 마이크로소프트는 2014년 10월 Windows 10의 시험판을 발표했음.
- 2015년 7월 출시.



Windows 10

Windows 8과 Window 10

■ Windows 10의 특징

- Windows 7과 Windows 8의 시작 버튼을 혼합시킨 새로운 형태의 시작 버튼
- 창 화면으로 작동되는 메트로 앱
- 새로운 데스크 뷰 버튼과 멀티 데스크탑
- 스냅 기능과 항상 빠른 파일 검색

구조	IA-32(32비트)	x86-64(64비트)
프로세서	1GHz 이상	
메모리(RAM)	1GB	2GB
그래픽카드	DirectX 9 그래픽 장치(WDDM 드라이버 포함)	
HDD	16GB 이상의 공간	
기타	Microsoft 계정과 인터넷	

Windows 10의 하드웨어 사양

■ UNIX의 역사

- 켄 톰슨과 데니스 리치가 MULTICS보다 더 작고 일관성 있는 운영체제를 목적으로 만든 것이 UNICS이고, 이것이 UNIX의 효시가 됨
- 1969년 : 벨 연구소에서 인터랙티브 시분할 시스템으로 만든 것이 시초가 된 OS임
- 1973~4년 : C 언어로 쓰여진 최초의 운영체계가 됨
- 여러 회사나 대학 및 개인들에 의해 많은 확장판과 새로운 아이디어들이 다양한 버전의 유닉스에 추가됨으로써, 대형 프리웨어 제품의 한 종류로 진화함



왼쪽부터 켄 톰슨, 데니스 리치

■ UNIX의 특징

- 전문 프로그래머용, 처음 배우기엔 어려운 경향이 있지만 익숙해지면 효율적으로 사용할 수 있음
- 고급언어로 작성되어 손쉬운 개발이 가능해짐
- 소스를 공개하여 UNIX의 발전을 꾀함
- 다중 사용자 시스템임
- UNIX는 C언어 외에도 기타 다른 언어, 그리고 프로그래밍에 필요한 도구들까지 기본으로 지원함
- 강력한 네트워크를 지원하고 X-Window라는 GUI환경 지원함

항목	장점	단점
이식성, 호환성, 공개성	개발에 용이함	다양한 버전이 있음
파일 시스템	트리 구조로 사용이 편리함	처리 속도 지연
시분할 방식	모든 사용자에게 컴퓨터 자원을 균등 분배	업무의 우선 순위 결정이 어려움
명령어 축약	최소로 최대의 효과	초보자는 사용이 어려움
통신	다양한 유틸리티 제공(UUCP, e-메일 등)	—
분산처리	서로 다른 기종 간에도 가능	—
프로그램 개발 도구	다양한 도구 제공	—
한글 지원	자국어 처리 시스템 이용	—
에러 메시지	—	간단하고 불분명함

UNIX의 장점과 단점

■ Linux의 역사

- 1991년 핀란드 헬싱키 대학 학생이던 리누스 토발스(Linus Torvalds)가 대형 기종에서나 작동하던 OS인 UNIX를 386 기종의 PC에서도 작동할 수 있게 만듦
- 인터넷을 통해 무료로 배포하고 있는 컴퓨터 운영체제임
- 프로그램 소스가 공개돼 있어 사용자가 원하는 대로 특정기능을 추가할 수 있고, 더욱이 어느 기종에도 사용이 가능함
- 현재에도 이러한 장점 때문에 일반 기업과 연구기관 등에서 수요가 늘어남
- 리눅스 커널(kernel)을 기반으로 한 GNU 시스템임

GNU란

GNU 프로젝트는 GNU 시스템이라고 불리는 유닉스 형태의 자유 소프트웨어 운영체제를 개발하기 위해 1984년부터 시작되었다.(GNU라는 단어는 “GNU’s Not UNIX”를 의미하는 재귀적 약어이며 ‘그-뉴’라고 발음한다.) 이 모임은 소프트웨어의 배포와 개발의 자유를 추구하는 그룹이며, Copyleft 운동으로도 잘 알려져 있다.

Linux의 개발자
리누스 토발즈



GNU 운동의 창시자
리처드 스톨만

■ Linux의 특징

- 강력한 성능의 완전한 공개 OS 프로그램임
- UNIX와 유사한 형태를 가지기 때문에 익히기가 쉬움
- 서버용 소프트웨어를 포함한 여러 소프트웨어를 기본으로 제공함
- 사용자가 사용하기 쉽도록 한 배포판을 무료로 다운 받을 수 있음

Linux



Linux 배포판들

■ Linux의 단점

- 책임지고 개발하는 사람들이 적음
- 현재도 개발되고 있는 운영체제이며, 업그레이드 등의 사후 관리가 어려움
- Linux는 컴퓨터에 대한 많은 지식을 요구함. 또한 자발적 정신으로 만들어지는 성격을 가지므로 단순한 문제는 사용자 스스로 해결해야 함

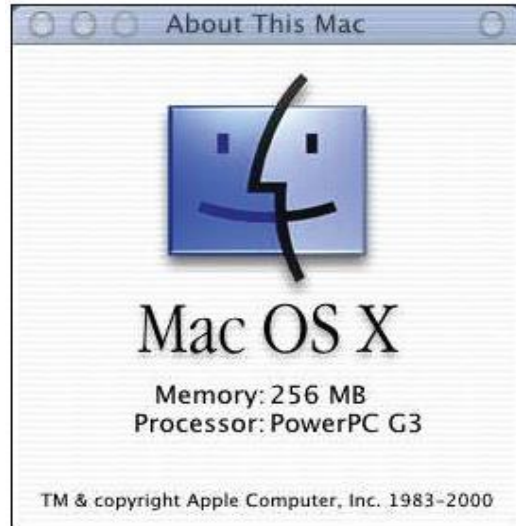
■ 매킨토시의 역사

- 매킨토시(Macintosh)는 줄여서 '맥'(Mac)이라고도 부름
- 애플컴퓨터사에서 만든 32 비트 개인용 컴퓨터임
- 모토로라 68000 계열의 CPU와 애플사 고유의 OS를 기반으로 하고 있음
- 매킨토시의 GUI개념은 1970년 초 제록스의 Parc 연구소에서 실험적으로 만들어진 것에 그 뿌리를 두고 있음
- 사용자의 입장을 가장 잘 반영한 OS 중의 하나로 처음부터 GUI를 바탕을 만들어 졌기 때문에 사용하기가 매우 쉬움

Mac OS



매킨토시 컴퓨터



Mac OS X의 화면과 box

Mac OS

■ Mac OS의 특징들

- 셸록(Sherlock) 기능 : 파일 찾기 명령과 비슷한 검색 기능. 인터넷 상에서 주요 디렉토리나 검색엔진을 찾은 후 그 결과를 마치 매킨토시 파일 시스템에서처럼 접근이 가능하게 함
- 다수사용자를 지원해 최대 40명까지 함께 사용할 수 있음
- 보안 기능 중 열쇠고리 기능을 제공해 여러 개의 사용자 아이디와 암호를 하나의 "키"로 관리함
- 인터넷을 통한 자동업데이트를 지원함
- 멀티미디어, 특히 2D 이미지나 소리 분야에서 다양한 기능을 지원함

Mac OS



Mac 바탕화면



Mac 폴더



셜록(sherlock)

Mac OS



맥 OS X → macOS

Windows 기본 명령어

윈도우 운영체제 기본 명령어

■ 윈도우 단축키

윈도우 단축키	단축키 기능	기타
윈도우 키 + E	윈도우 탐색기 실행	'내 컴퓨터' 와 동일
윈도우 키 + F	윈도우 검색창 실행	'시작 - 검색' 과 동일
윈도우 키 + R	윈도우 실행창 실행	'시작 - 실행' 과 동일
윈도우 키 + D	모든 창 최소화	다시 단축키 누르면 창이 원래대로 돌아옴
윈도우 키 + M	모든 창을 최소화	다시 단축키를 눌러도 창이 최대화되지 않음
윈도우 키 + Shift + M	최소화한 창을 최대화	'윈도우 키 + M' 단축키의 반대 동작
윈도우 키 + L	컴퓨터 잠금	잠금 해제시 사용지 비밀번호 필요
윈도우 키 + Pause/Break	시스템 등록정보 창을 띄움	'시작 - 제어판 - 시스템' 과 동일

윈도우 운영체제 기본 명령어

■ 실행창 기본명령어 - '시작-실행' (단축키 : 윈도우즈키 + R)후 입력

control : 제어판

Access.cpl : 내게 필요한 옵션

appwiz.cpl : 프로그램 추가/제거

bthprops.cpl : 블루투스장치설정

desk.cpl : 디스플레이 등록정보

firewall.cpl : Windows방화벽

hdwwiz.cpl : 새하드웨어추가마법사

inetcpl.cpl : 인터넷등록정보

intl.cpl : 국가및언어옵션

irprops.cpl : 적외선포트 설정

joy.cpl : 게임컨트롤러

main.cpl : 마우스등록정보

mmsys.cpl : 사운드 및 오디오장치등록정보

ncpa.cpl : 네트워크연결

netsetup.cpl : 네트워크설정마법사

nusrmgr.cpl : 사용자계정

nwc.cpl : 네트워크 게이트웨이

odbc32.cpl : ODBC데이터원본 관리자

powercfg.cpl : 전원옵션 등록정보

sysdm.cpl : 시스템등록정보

telephon.cpl : 전화및모뎀 옵션

timedate.cpl : 날짜 및 시간 등록정보

wscui.cpl : Windows보안센터

wuauclt.cpl : 자동업데이트

Sapi.cpl : 텍스트 음성 변환설정

윈도우 운영체제 기본 명령어

■ 관리콘솔 명령어

certmgr.msc : 인증서

ciadv.msc : 인덱싱서비스

ntsmgr.msc : 이동식저장소

ntmsoprq.msc : 이동식저장소 운영자 요청

secpol.msc : 로컬보안정책

wmimgmt.msc : WMI

(Windows Management Infrastructure)

compmgmt.msc : 컴퓨터 관리

devmgmt.msc : 장치관리자

diskmgmt.msc : 디스크 관리

dfrg.msc : 디스크 조각모음

eventvwr.msc : 이벤트 뷰어

fsmgmt.msc : 공유폴더

gpedit.msc : 로컬 컴퓨터 정책

lusrmgr.msc : 로컬 사용자 및 그룹

perfmon.msc : 성능모니터뷰

rsop.msc : 정책의 결과와 집합

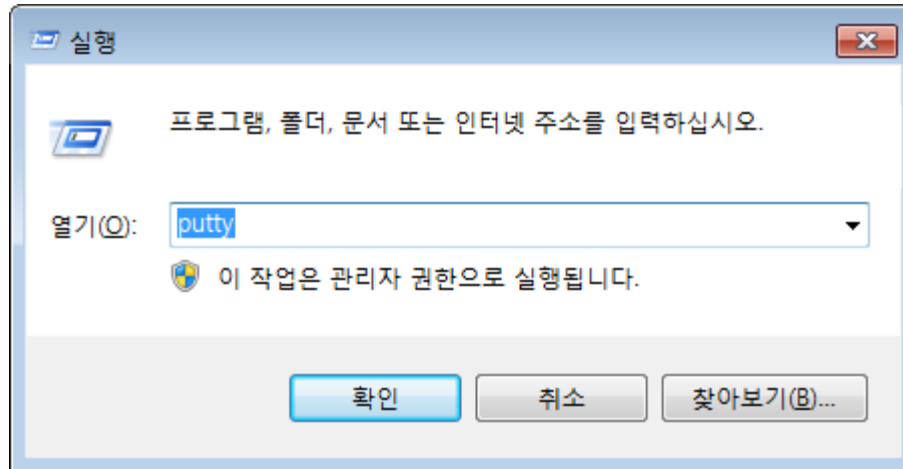
secpol.msc : 로컬 보안설정

services.msc : 서비스

윈도우 운영체제 기본 명령어

■ 활용

- 단순 실행 파일이라면 C:\Windows\System32 폴더에 집어 넣으면, 시작 - 실행에 해당 파일 이름을 실행시키는 것으로 편하게 사용할 수 있습니다.
- ex) Putty.exe 파일을 System32 폴더에 집어넣고. 이런식으로 사용 가능합니다.



윈도우 운영체제 기본 명령어

■ CMD 명령어

명령어	설명	설명
1	<u>ASSOC</u>	파일 확장명 연결을 보여주거나 수정합니다.
2	<u>ATTRIB</u>	파일 속성을 표시하거나 바꿉니다.
3	<u>BREAK</u>	확장된 CTRL+C 검사를 설정하거나 지웁니다.
4	<u>BCDEDIT</u>	부팅 로딩을 제어하기 위해 부팅 데이터베이스에서 속성을 설정합니다.
5	<u>CACLS</u>	파일의 액세스 컨트롤 목록(ACL)을 표시하거나 수정합니다.
6	<u>CALL</u>	한 일괄 프로그램에서 다른 일괄 프로그램을 호출합니다.
7	<u>CD</u>	현재 디렉터리 이름을 보여주거나 바꿉니다.
8	<u>CHCP</u>	활성화된 코드 페이지의 번호를 표시하거나 설정합니다.
9	<u>CHDIR</u>	현재 디렉터리 이름을 보여주거나 바꿉니다.
10	<u>CHKDSK</u>	디스크를 검사하고 상태 보고서를 표시합니다.
11	<u>CHKNTFS</u>	부팅하는 동안 디스크 확인을 화면에 표시하거나 변경합니다.
12	<u>CLS</u>	CMD 화면에 표시된 것을 모두 지웁니다.
13	<u>CMD</u>	Windows 명령 인터프리터의 새 인스턴스를 시작합니다.
14	<u>COLOR</u>	콘솔의 기본색과 배경색을 설정합니다.
15	<u>COMP</u>	두 개 또는 여러 개의 파일을 비교합니다.
16	<u>COMPACT</u>	NTFS 분할 영역에 있는 파일의 압축을 표시하거나 변경합니다.
17	<u>CONVERT</u>	FAT 볼륨을 NTFS로 변환합니다. 현재 드라이브는 변환할 수 없습니다.
18	<u>COPY</u>	하나 이상의 파일을 다른 위치로 복사합니다.
19	<u>DATE</u>	날짜를 보여주거나 설정합니다.
20	<u>DEL</u>	하나 이상의 파일을 지웁니다.
21	<u>DIR</u>	디렉터리에 있는 파일과 하위 디렉터리 목록을 보여줍니다.
22	<u>DISKCOMP</u>	두 플로피 디스크의 내용을 비교합니다.
23	<u>DISKCOPY</u>	플로피 디스크의 내용을 다른 플로피 디스크로 복사합니다.
24	<u>DISKPART</u>	디스크 파티션 속성을 표시하거나 구성합니다.

윈도우 운영체제 기본 명령어

■ CMD 명령어

명령어	설명	설명
25	<u>DOSKEY</u>	명령줄을 편집하고, Windows 명령을 다시 호출하고, 매크로를 만듭니다.
26	<u>DRIVERQUERY</u>	현재 장치 드라이버 상태와 속성을 표시합니다.
27	<u>ECHO</u>	메시지를 표시하거나 ECHO를 사용 또는 사용하지 않습니다.
28	<u>ENDLOCAL</u>	배치 파일에서 환경 변경의 지역화를 끝냅니다.
29	<u>ERASE</u>	하나 이상의 파일을 지웁니다.
30	<u>EXIT</u>	CMD.EXE 프로그램(명령 인터프리터)을 마칩니다.
31	<u>FC</u>	두 파일 또는 파일의 집합을 비교하여 다른 점을 표시합니다.
32	<u>FIND</u>	파일에서 텍스트 문자열을 찾습니다.
33	<u>FINDSTR</u>	파일에서 문자열을 찾습니다.
34	<u>FOR</u>	파일 집합에서 각 파일에 대해 지정된 명령을 실행합니다.
35	<u>FORMAT</u>	Windows에서 사용할 디스크를 포맷합니다.
36	<u>FSUTIL</u>	파일 시스템 속성을 표시하거나 구성합니다.
37	<u>FTYPE</u>	파일 확장 연결에 사용되는 파일 형식을 표시하거나 수정합니다.
38	<u>GOTO</u>	Windows 명령 인터프리터가 일괄 프로그램에서 이름표가 붙여진 줄로 이동합니다.
39	<u>GPRESULT</u>	컴퓨터 또는 사용자에게 대한 그룹 정책 정보를 표시합니다.
40	<u>GRAFTABL</u>	Windows가 그래픽 모드에서 확장 문자 세트를 표시할 수 있게 합니다.
41	<u>HELP</u>	Windows 명령어에 관한 도움말을 제공합니다.
42	<u>ICACLS</u>	파일과 디렉터리에 대한 ACL을 표시, 수정, 백업 또는 복원합니다.
43	<u>IF</u>	일괄 프로그램에서 조건 처리를 수행합니다.
44	<u>LABEL</u>	디스크의 볼륨 이름을 만들거나, 바꾸거나, 지웁니다.
45	<u>MD</u>	디렉터리를 만듭니다.
46	<u>MKDIR</u>	디렉터리를 만듭니다.
47	<u>MKLINK</u>	기호화된 링크와 하드 링크를 만듭니다.
48	<u>MODE</u>	시스템 장치를 구성합니다.

윈도우 운영체제 기본 명령어

■ CMD 명령어

명령어	설명	설명
49	MORE	출력을 한번에 한 화면씩 표시합니다.
50	MOVE	하나 이상의 파일을 한 디렉터리에서 다른 디렉터리로 옮깁니다.
51	OPENFILES	파일 공유에서 원격 사용자에게 의해 열린 파일을 표시합니다.
52	PATH	실행 파일의 찾기 경로를 보여주거나 설정합니다.
53	PAUSE	배치 파일의 처리를 보류하고 메시지를 보여줍니다.
54	POPD	PUSHD 명령으로 저장된 디렉터리로 바꿉니다.
55	PRINT	텍스트 파일을 인쇄합니다.
56	PROMPT	Windows 명령 프롬프트를 바꿉니다.
57	PUSHD	현재 디렉터를 저장한 후 디렉터를 바꿉니다.
58	RD	디렉터를 지웁니다.
59	RECOVER	불량이거나 결함이 있는 디스크에서 읽을 수 있는 정보를 복구합니다.
60	REM	배치 파일 또는 CONFIG.SYS에서 주석을 기록합니다.
61	REN	파일 이름을 바꿉니다.
62	RENAME	파일 이름을 바꿉니다.
63	REPLACE	파일을 대체합니다.
64	RMDIR	디렉터를 지웁니다.
65	ROBOCOPY	파일과 디렉터리 트리를 복사할 수 있는 고급 유틸리티입니다.
66	SET	Windows 환경 변수를 보여주거나, 설정하거나, 지웁니다.
67	SETLOCAL	배치 파일에서 환경 변경의 지역화를 시작합니다.
68	SC	서비스를 표시하거나 구성합니다(백그라운드 작업).
69	SCHTASKS	컴퓨터에서 실행할 명령과 프로그램을 예약합니다.
70	SHIFT	배치 파일에서 바꿀 수 있는 매개 변수의 위치를 바꿉니다.
71	SHUTDOWN	컴퓨터의 로컬 또는 원격 종료를 허용합니다.
72	SORT	입력을 정렬합니다.

윈도우 운영체제 기본 명령어

■ CMD 명령어

명령어	설명	설명
74	<u>SUBST</u>	경로를 드라이브 문자로 지정합니다.
75	<u>SYSTEMINFO</u>	특정 컴퓨터의 속성과 구성을 표시합니다.
76	<u>TASKLIST</u>	서비스를 포함한 실행 중인 모든 작업을 표시합니다.
77	<u>TASKKILL</u>	실행 중인 프로세스나 응용 프로그램을 중지합니다.
78	<u>TIME</u>	시스템 시간을 보여주거나 설정합니다.
79	<u>TITLE</u>	CMD.EXE 세션에 대한 창의 창 제목을 설정합니다.
80	<u>TREE</u>	드라이브 또는 경로의 디렉터리 구조를 그래픽으로 화면에 표시합니다.
81	<u>TYPE</u>	텍스트 파일의 내용을 보여줍니다.
82	<u>VER</u>	Windows 버전을 보여줍니다.
83	<u>VERIFY</u>	파일이 디스크에 올바르게 쓰였는지 검증할지 여부를 지정합니다.
84	<u>VOL</u>	디스크 볼륨 이름과 일련 번호를 보여줍니다.
85	<u>XCOPY</u>	파일과 디렉터리 트리를 복사합니다.
86	<u>WMIC</u>	대화형 명령 셸 내의 WMI 정보를 표시합니다.
87	<u>START</u>	해당 프로그램을 실행합니다.

Unix / Linux

동기

■ 유닉스/리눅스 운영체제

- 1970년대 초에 AT&T 벨연구소에서 개발된 이후로 지속적으로 발전
- 스마트폰, PC, 서버 시스템, 슈퍼컴퓨터에까지 사용되고 있음
- 소프트웨어 경쟁력의 핵심이 되고 있다.

■ 유닉스/리눅스 기반 운영체제

- 안드로이드(Android) OS
- iOS
- 맥(Mac) OS X
- 4. 리눅스(Linux)
- 5. BSD 유닉스(Unix)
- 6. 시스템 V
- 7. Sun 솔라리스(Solaris)
- 8. IBM AIX
- 9. HP HP-UX
- 10. Cray 유니코스(Unicos)

유닉스의 설계 철학

■ 단순성

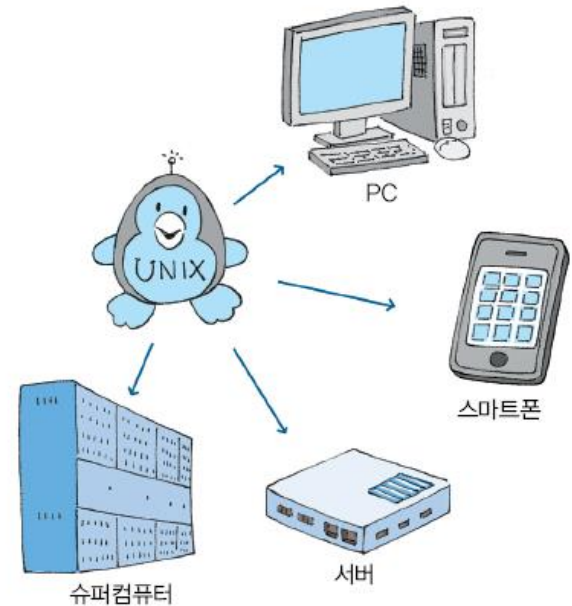
- MIT MULTICS에 반대해서 최소한의 기능만 제공
- 자원에 대한 일관된 관점 제공

■ 이식성

- 이식성을 위해 C 언어로 작성
- 다양한 플랫폼에 이식 가능
- 스마트폰, PC, 서버, 슈퍼컴퓨터 등

■ 개방성

- 소스 코드 공개와 같은 개방성



유닉스의 특징

■ 다중 사용자, 다중 프로세스 운영체제

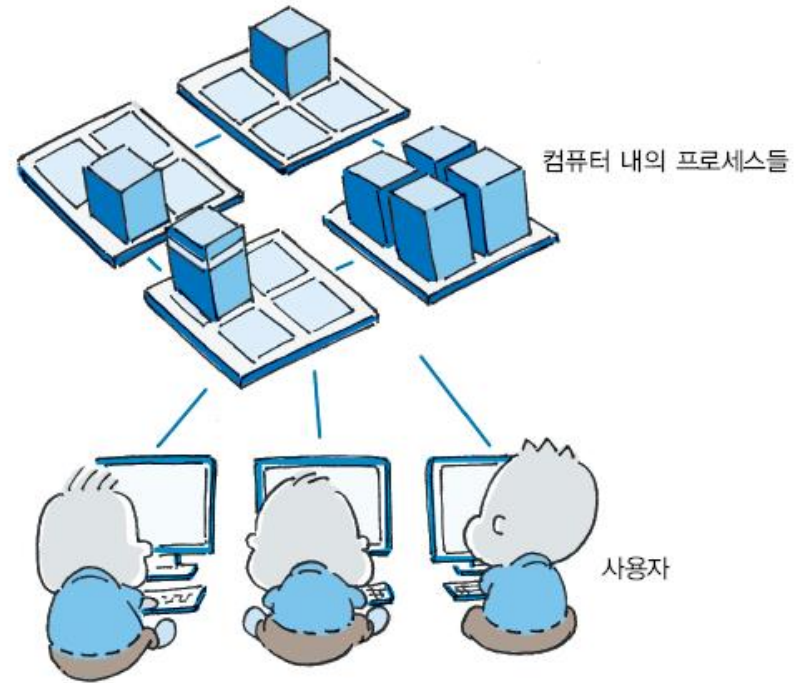
- 여러 사용자가 동시에 사용 가능
- 여러 프로그램이 동시에 실행
- 관리자 슈퍼유저가 있음.

■ 셸 프로그래밍

- 명령어나 유틸리티 등을 사용하여 작성한 프로그램

■ 훌륭한 네트워킹

- 유닉스에서부터 네트워킹이 시작
- ftp, telnet, WWW, X-window 등



유닉스 운영체제 구조

■ 운영체제

- 컴퓨터의 하드웨어 자원을 운영 관리하고
- 프로그램을 실행할 수 있는 환경을 제공.

■ 커널(kernel)

- 운영체제의 핵심으로 하드웨어 운영 및 관

■ 시스템 호출(system call)

- 커널이 제공하는 서비스에 대한
- 프로그래밍 인터페이스 역할

■ 셸(shell)

- 사용자와 운영체제 사이의 인터페이스
- 사용자로부터 명령어를 입력 받아
- 해석하여 수행해주는 명령어 해석기

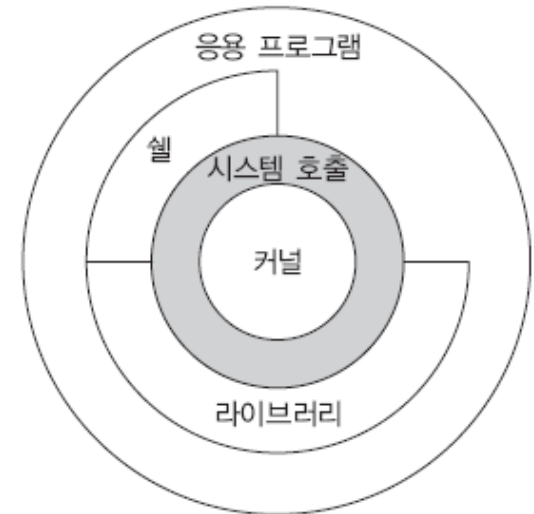
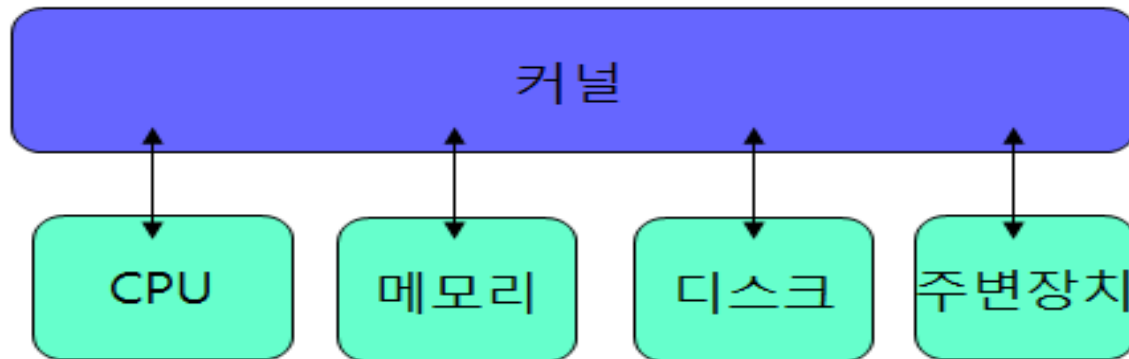


그림 1.3 유닉스 운영체제 구조

커널

■ 커널의 역할

- 하드웨어를 운영 관리하여
- 프로세스, 파일, 메모리, 통신, 주변장치 등을
- 관리하는 서비스를 제공한다.



커널의 역할

■ 프로세스 관리(Process management)

- 여러 프로그램이 실행될 수 있도록
- 프로세스들을 CPU 스케줄링하여 동시에 수행되도록 한다.

■ 파일 관리(File management)

- 디스크와 같은 저장장치 상에 파일 시스템을 구성하여 파일을 관리한다.

■ 메모리 관리(Memory management)

- 메인 메모리가 효과적으로 사용될 수 있도록 관리한다.

■ 통신 관리(Communication management)

- 네트워크를 통해 정보를 주고받을 수 있도록 관리한다.

■ 주변장치 관리(Device management)

- 모니터, 키보드, 마우스와 같은 장치를 사용할 수 있도록 관리한다.

유닉스 역사 및 표준

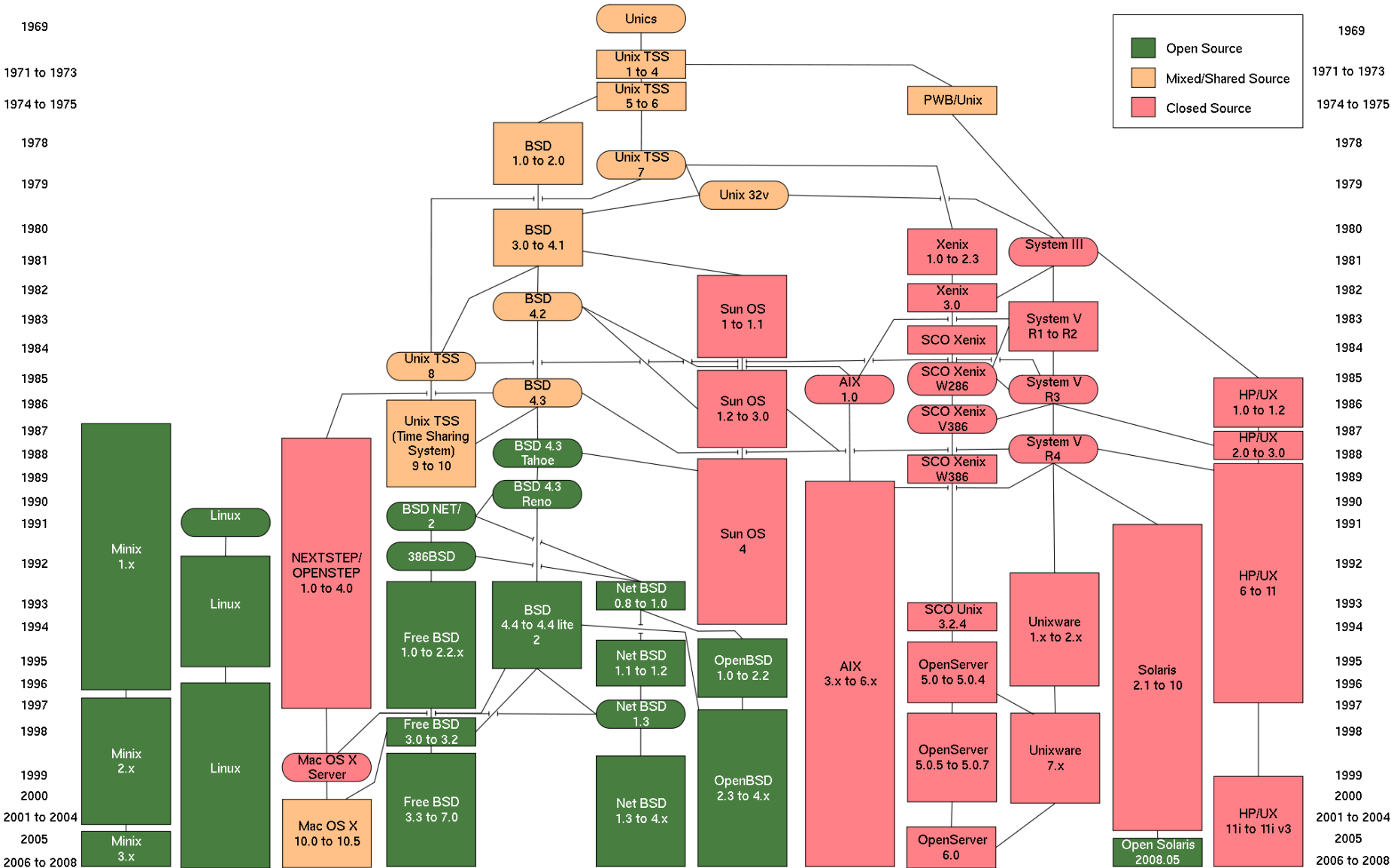
● AT&T 벨 연구소(Bell Lab)에서 개발됨

- Ken Thompson이 어셈블리어로 개발함
- D. Ritchie가 C 언어로 다시 작성함
 - C 언어는 Unix를 작성하기 위한 언어로 밀접하게 관련되어 있음
- 이론적으로 C 컴파일러만 있으면 이식 가능
- 소스 코드를 대학에 개방함

● 유닉스의 큰 흐름

- 시스템 V(System V)
- BSD(Berkeley Standard Distribution) 유닉스
- 리눅스(Linux)

유닉스 버전 트리[위키백과]



유닉스 시스템 V

- 벨 연구소에서 개발된 버전이 발전하여 시스템 V가 됨
- 유닉스 버전 중의 최초의 대표적인 성공 사례
 - 여러 유틸리티가 공개되면서 일반 사용자에게 확산
- 다양한 상업용 버전으로 발전
 - IBM의 AIX, Sun의 Solaris, HP의 UP-UX



BSD 유닉스

- **공개 소스코드를 기반으로 버클리대학교에서 개선**

- 지속적으로 발전하여 BSD 4.3 버전이 개발됨

- **주요 기능 개선**

- 메모리 관리 기능 향상
- 네트워킹 기능 추가
 - TCP/IP 네트워킹, 소켓(Socket) 등

- **상업용 운영체제의 기초**

- 썬 OS(Sun OS), 맥 OS(Mac OS) 등

리눅스

- **PC를 위한 효율적인 유닉스 시스템**

- 1991년 헬싱키 대학의 Linus B. Torvalds에 의해 개발됨

- **소스코드가 공개**

- 인터넷 상에서 자원자들에 의해서 기능 추가 및 확장됨
- 공용 도메인 상의 무료 OS

- **다양한 플랫폼에 포팅 가능**

- PC, 워크스테이션, 서버 등

- **GNU 소프트웨어와 함께 배포**

- GNU/Linux 운영체제



리눅스 장점

- **풍부하고 다양한 하드웨어를 효과적으로 지원**

- 대부분의 하드웨어를 지원하는 추세임
- PC, 워크스테이션, 서버 등

- **놀라운 성능 및 안정성**

- Pentium으로도 충분히 빠르며 안전하게 수행

- **인터넷에 맞는 강력한 네트워크 구축**

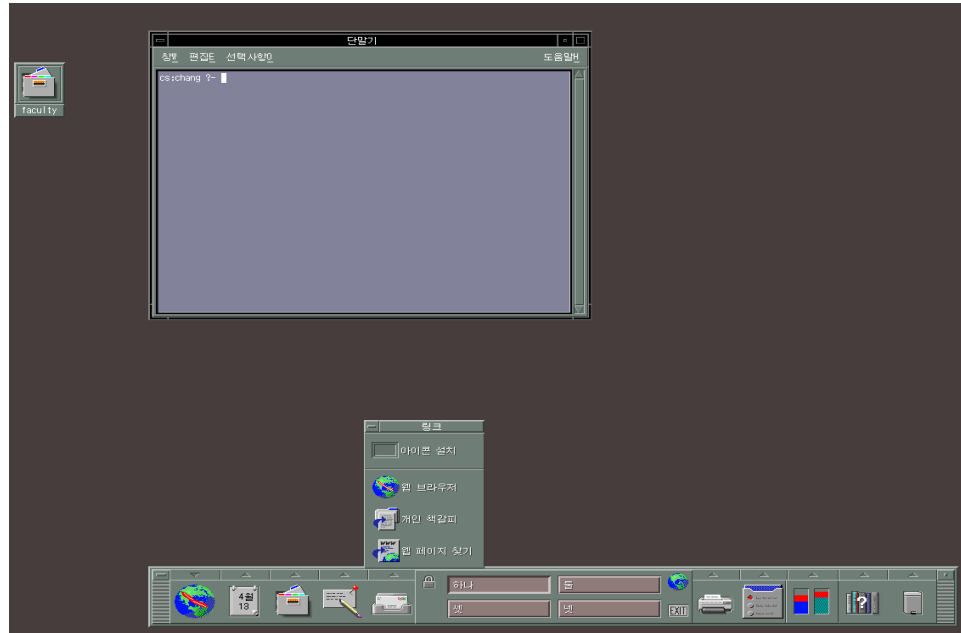
- **다양한 응용 프로그램 개발됨**

- **무료 배포판**

- 레드햇(RedHat): 상업용
- 우분투(Ubuntu)
- 페도라(Fedora)
- CentOS

솔라리스(Solaris)

- 썬(SUN)에서 개발한 시스템 V 기반의 운영체제
 - 썬 워크스테이션에서 전문가들이 주로 사용



맥 OS(Mac OS)

● 1984년 애플 매킨토시 컴퓨터용 운영체제로 개발

- 개인용 컴퓨터에 GUI를 처음으로 도입

● 맥 OS X

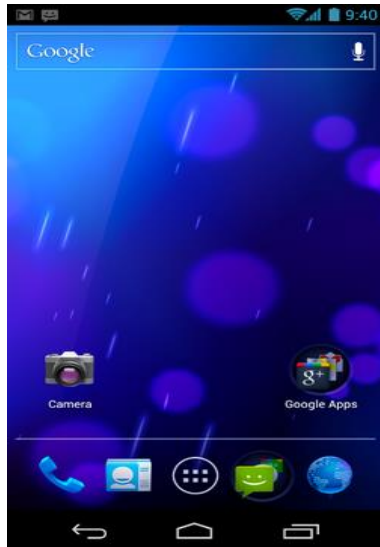
- 2002년에 NeXTSTEP 운영체제와 BSD 유닉스를 기반으로 개발
- 문서편집, 그래픽, 멀티미디어 등의 분야에서 많이 사용됨



모바일 기기용 운영체제

■ 안드로이드(Android)

- 리눅스 기반 모바일 기기용
- 주로 스마트폰, 태블릿 PC 등
- 개방형 운영체제로 소스 코드 등 공개



■ iOS

- 맥 OS X를 기반으로 개발된 모바일 기기용 운영체제
- 애플사의 iPhone, iPad, iPod



리눅스 설치

■ 다양한 배포판

- 커널은 공유하고 배포판마다 조금씩 다른 데스크톱 환경이나 응용 프로그램 제공
- 레드햇(RedHat) : 상업용
- 우분투(Ubuntu), CentOS, 데비안(Debian), 페도라(Fedora) 등

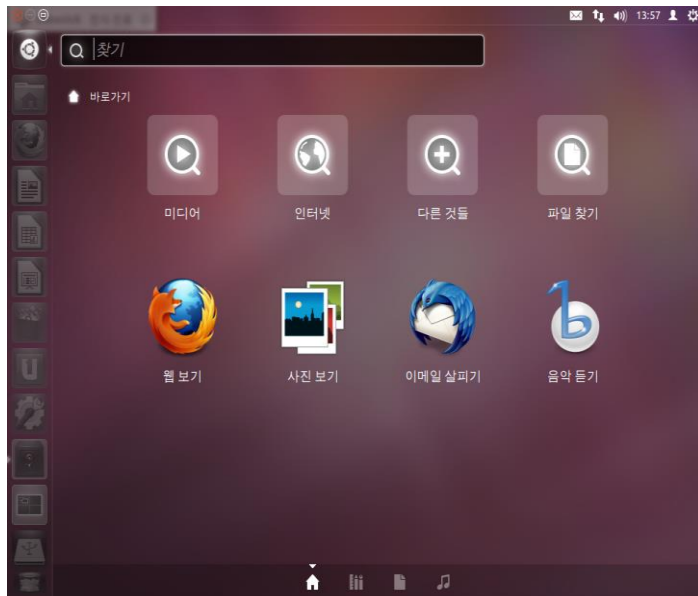
■ 데스크톱 환경

- 사용하는 데스크톱 환경에 따라 사용방법이나 사용 가능한 응용 프로그램이 조금씩 다르다.
- GNOME, KDE, Unity

리눅스 배포판

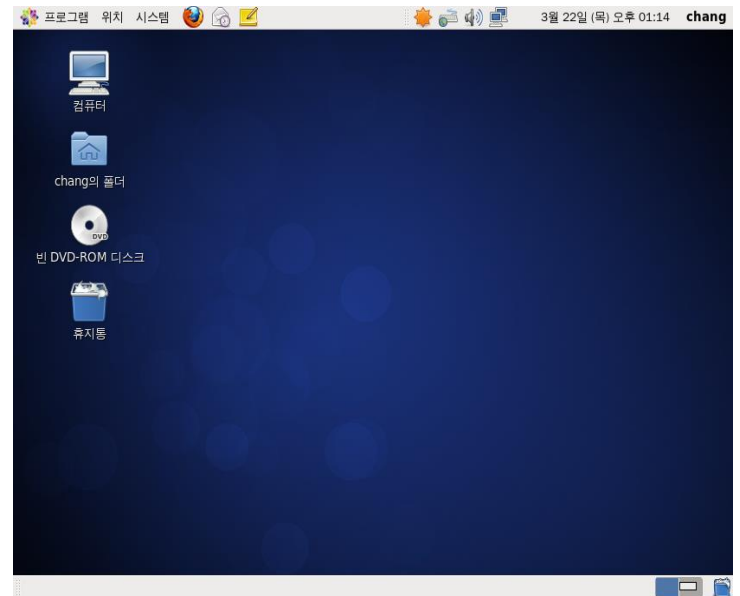
■ 우분투(Ubuntu)

- 데스크톱에서 쉽게 사용할 수 있는 리눅스 배포판
- MS 윈도우즈 상에서 인터넷으로 쉽게 직접 설치도 가능
- <http://www.ubuntu.com>



■ CentOS

- RedHat Enterprise 배포판을 기반으로 하는 무료 운영체제
- 주로 서버용으로 많이 사용되며
- 데스크톱용, 워크스테이션용도 제공
- <http://www.centos.org>



우분투 설치

■ 우분투 데스크탑 다운로드

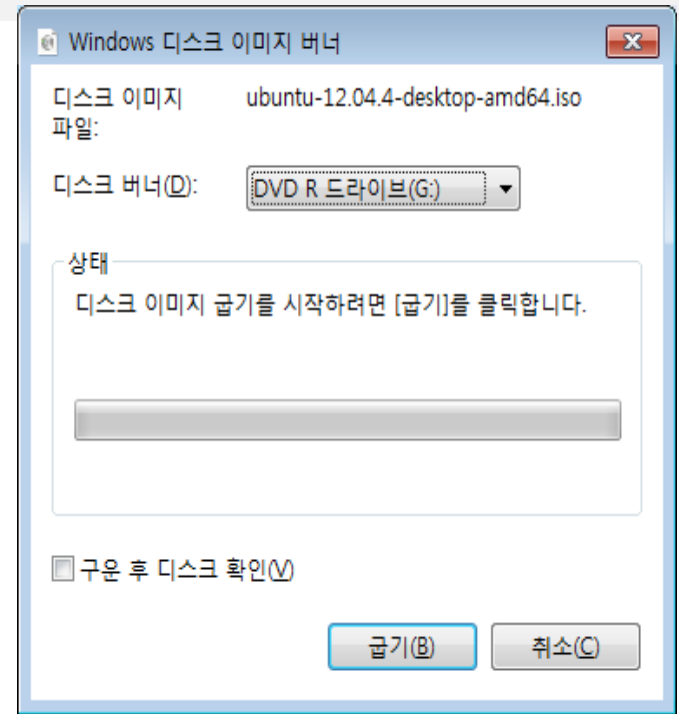
- <http://www.ubuntu.com/download/desktop>

■ 디스크 굽기(Burning)

- 배포판 파일을 빈 DVD에 복사하는 과정
- 디스크 이미지 버너(그림 1.10) 이용
- 배포판 파일을 다운받아 클릭하면 시작

■ 우분투 데스크톱 설치

- Install Ubuntu alongside Windows 7 혹은 Replace Windows 7 with Ubuntu 선택



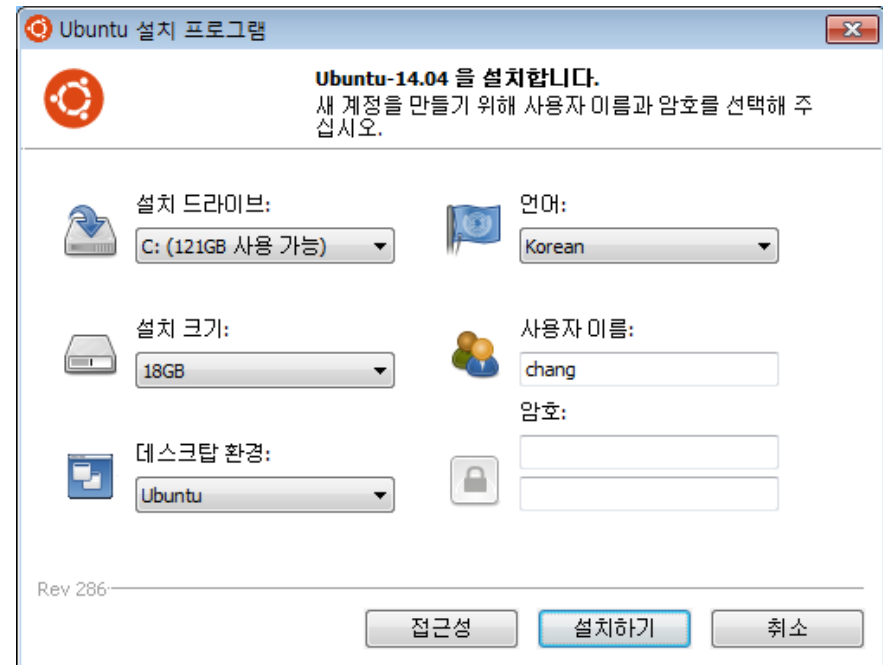
Wubi를 이용한 우분투 설치

■ DVD 설치 디스크 없이 설치하는 간단한 방법

- Wubi 이용하면 거의 모든 설치 과정이 자동

■ Wubi(Window-based Ubuntu Installer)

- 기존의 MS 윈도우즈 운영체제를 그대로 유지하면서
- 우분투 리눅스를 새로 설치할 수 있다.
- 원하는 배포판의 Wubi를 다운로드
- <http://releases.ubuntu.com>



레드햇 리눅스와 CentOS

■ 레드햇 리눅스(Red Hat Linux)

- 레드햇사가 개발한 리눅스 배포판
- 레드햇 엔터프라이즈 리눅스(RHEL)
 - 기업용 엔터프라이즈 컴퓨팅 플랫폼을 제공하는 유료 배포판

■ CentOS(Community ENTerprise Operating System)

- 레드햇 엔터프라이즈 배포판(RHEL) 기반의 무료 운영체제
- 웹 서버용, 데스크톱용, 워크스테이션용 등도 제공

CentOS 설치

● CentOS 설치

- 배포판을 다운받아 DVD 형태로 구운 후에 설치할 수 있다.
- CentOS 홈페이지 <http://www.centos.org>
- 국내 미리 사이트 http://ftp.daum.net/centos/6.5/isos/x86_64/

-

● 배포판

- 완전 버전(권장) [CentOS-6.5-x86_64-bin-DVD.iso](#)
- 최소 버전 [CentOS-6.5-x86_64-minimal.iso](#)
- 네트워크 설치 버전 [CentOS-6.5-x86_64-netinstall.iso](#)

-

● 완전 버전 설치할 때 옵션

- 서버용, 데스크톱용, 워크스테이션용, 최소용 등 선택 가능
- 자세한 설치 과정

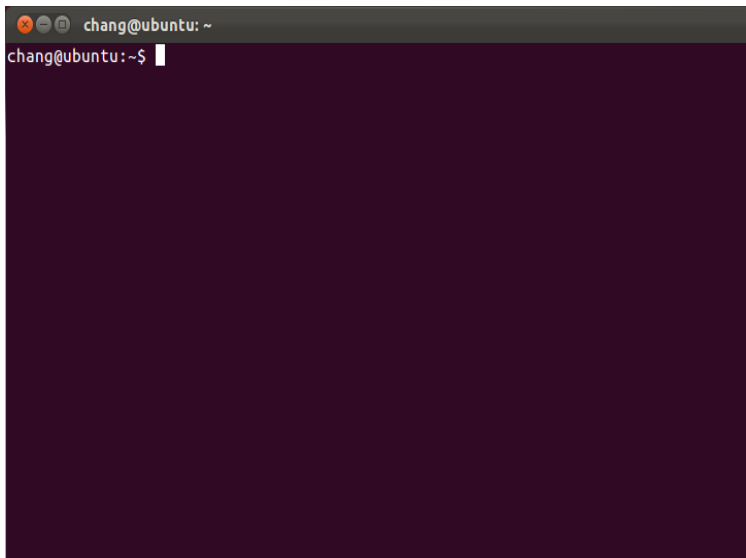
<http://linuxmoz.com/how-to-install-centos-6-linux-for-servers-desktops>

직접 로그인

■ 사용 가능한 유닉스/리눅스 시스템이 있는 경우

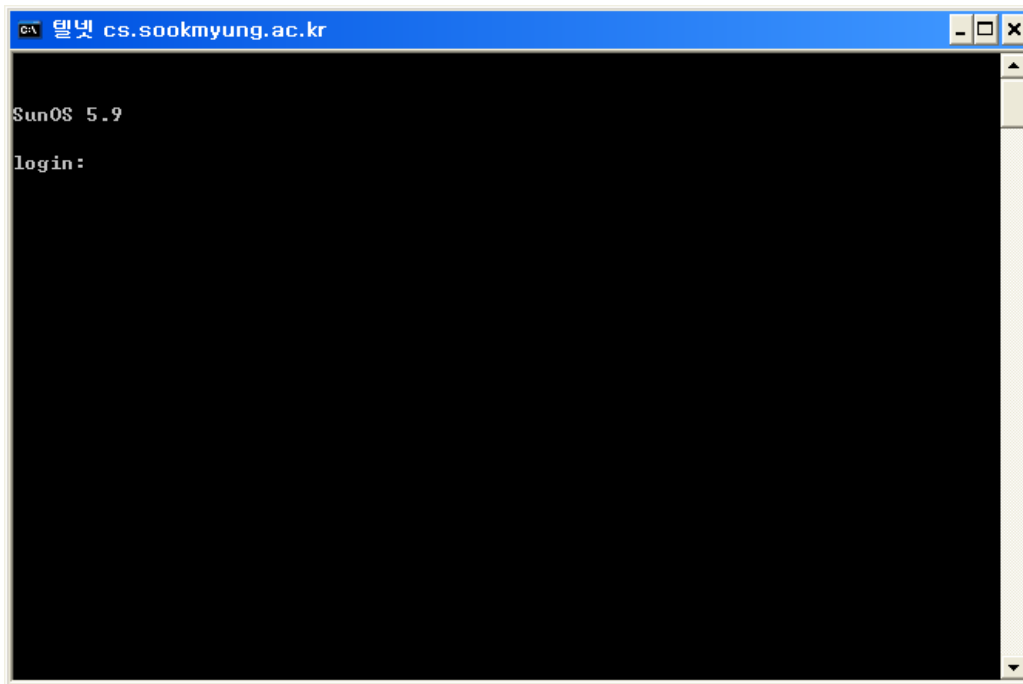
- X-윈도우(X-window)로 직접 로그인하여
- 바로 X-윈도우 시스템을 사용할 수 있다.

■ 우분투/CentOS 터미널 화면



원격 로그인

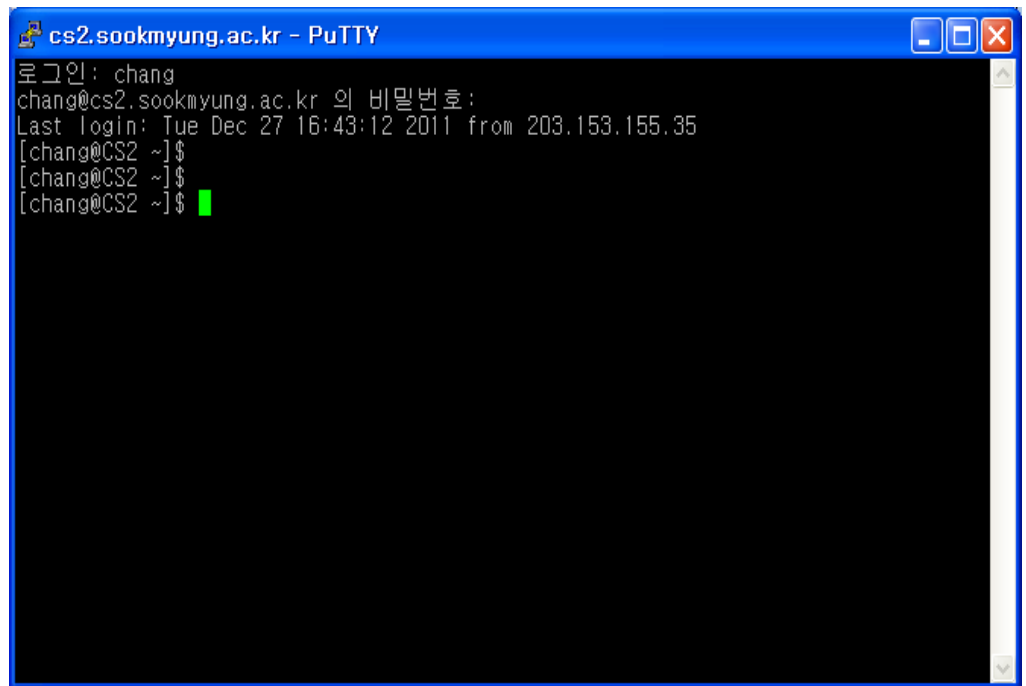
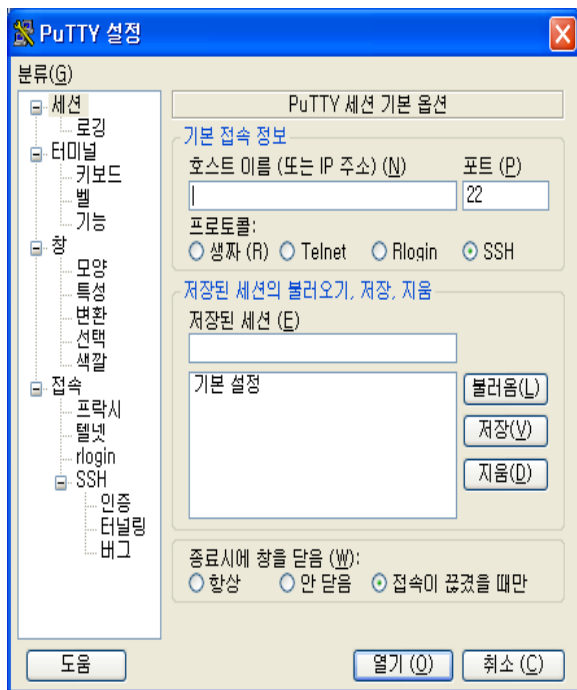
- MS 윈도우에서 telnet 이용



원격 로그인

■ PuTTY 사용

- <http://www.chiark.greenend.org.uk/~sgtatham/putty>
- telnet, ssh 등 이용하여 원격 로그인



시스템 관리자

■ 슈퍼유저(superuser)

- 시스템을 관리할 수 있는 사용자로
- 슈퍼유저가 사용하는 계정이 root이다

■ 슈퍼유저 로그인

- 직접 root 계정으로 로그인
- 다른 계정으로 로그인 후
- `$ su [사용자명]`

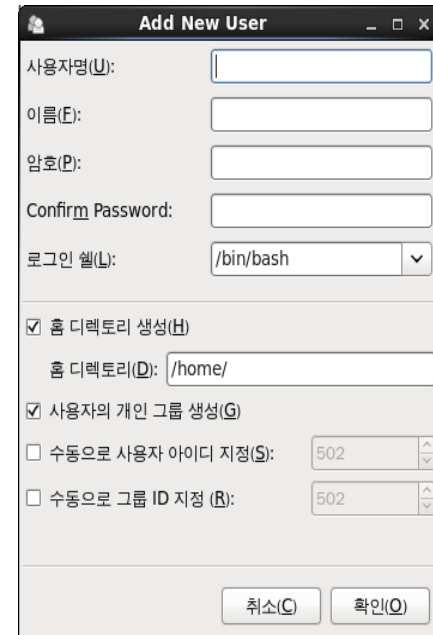
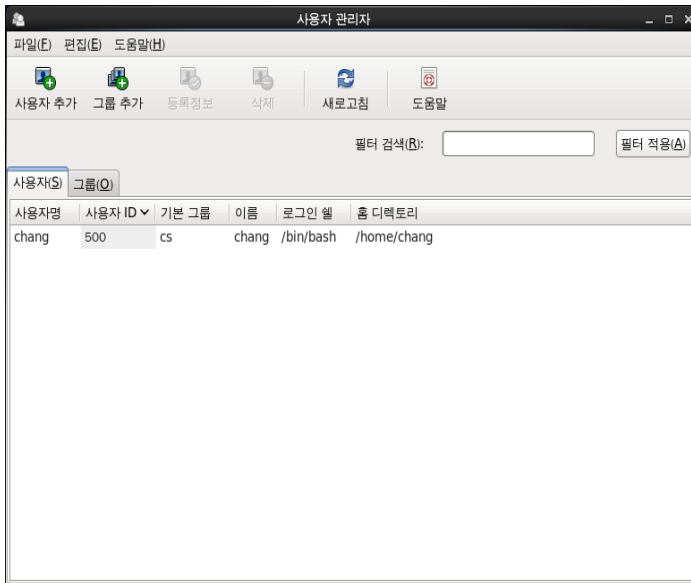
사용자 계정 추가

■ 사용자 추가/삭제

- # useradd [옵션] 사용자명 # userdel 사용자명
- # passwd 사용자명
- 관련 파일: /etc/passwd, /etc/shadow

■ 사용자 관리자 도구

- [시스템] -> [관리] -> [사용자 및 그룹]



그룹 추가

● 그룹 추가/삭제

- # groupadd [-g gid] 그룹명
- # groupdel 그룹명

Shell

■ 알아야 할 환경 변수

- `$>env` 입력하면 현재 내가 사용하고 있는 shell의 환경변수가 출력됨
 - `$SHELL` : 내가 현재 사용하고 있는 쉘
 - `$PATH` : 명령어의 path 정보 앞에서부터 차례로 입력
 - `$LANG` , `$LC_ALL`: 내 쉘의 char-set 정보 (`$LANG` < `$LC_ALL`이 우선순위 높음)
 - ex) `alias utf8='export LC_ALL=ko_KR.utf8'`

■ 알아야 할 쉘 시작(기동실행) 파일

쉘 마다 다름

bash : `.bash_profile` , `.bashrc`

csh : `.cshrc`

korn : `.profile`

Linux 주요 Directory

■ /etc

- 리눅스 운영 / 응용프로그램등의 환경 정보 파일이 모여있음
- /etc/passwd , /etc/hosts , /etc/xinetd.d

■ /bin , /usr/bin , /sbin , /usr/sbin

- 리눅스 명령어 모음

■ /var

- 리눅스 데이터 모음(로그, 메일등...)
- /var/log/secure , /var/log/ messages

■ /home

- 일반 사용자 홈 디렉터리

■ /lib

- C 라이브러리 모음

■ /tmp

- 임시디렉터리 (일반사용자로 Write가능)

Linux 주요 Command – 알아야 할 명령어들

- **ls** (list)
 - 파일 리스트 출력
 - 수많은 옵션이 존재 `$>ls -ltr`
 - Ex) `drwxrwxr-x 5 ir ir 4096 10월 19 11:24 abc`
 - `-rw-rw-r-- 1 root ir 203 12월 23 2009 test.sql`
- **cd** (change directory)
 - 디렉터리 이동 상위 `../`
 - 상대 path `../dir`
 - 절대 path `/dir`
 - `cd ~` : 사용자의 홈디렉터리 이동

Linux 주요 Command – 알아야 할 명령어들

- **cp (copy)**

파일 복사

옵션 -r , -R 디렉토리 전체 복사

- **mv(move)**

- `$>mv <source> <target>`

파일 이동 , 디렉토리 모두 가능

- **man (manual)**

- 명령 매뉴얼 보기 `$> man rm`

- **rm (remove)**

파일 삭제

옵션 : -r 하위 디렉토리 포함 삭제 , -f 강제 삭제

- **rmdir** : 디렉토리만 삭제 `<-> mkdir`

- **cat (concatenate)**

- 파일 내용출력

- **echo**

- 스트링또는 변수 값을 출력 `$>echo $SHELL`

- **sleep** 초단위 sleep

- **uname -a** 서버 네임 (cpu bit , os name)

Linux 주요 Command – 알아야 할 명령어들

- **touch** : 빈파일생성
- **chmod** : 파일의 권한 설정: 숫자 설정방식이 유용함 , -R 옵션은 하위 포함
U(user) , G(group) , O(other)
rwx : read(4) , write(2) , excute(1)
예제) 파일 소유자는 읽기와 실행권한 , 그룹에는 쓰기권한 , Other 에는 권한 없음
\$> chmod 520 <testfile>
- **chown** : 파일의 소유권 설정
 - 모든 파일에는 소유자와 그룹값이있음 , -R 옵션은 하위 포함예제 실행 쉘이 root권한일때 test.sql 파일의 소유자를 root로 변경
\$> chown root:irteam test.sql
- **pwd** (print working directroy) : 현재 디렉터리 출력
- **whereis , which**
 - 명령어의 위치를 찾아줌
- **w, who** : 현재 서버에 접속된 사용자 정보
 - 접속시간 , sorce ip , idle 타임...
- **uptime**
 - 서버의 부팅후 수행시간
 - cpu Load 정보

Vi(Visual) Editor

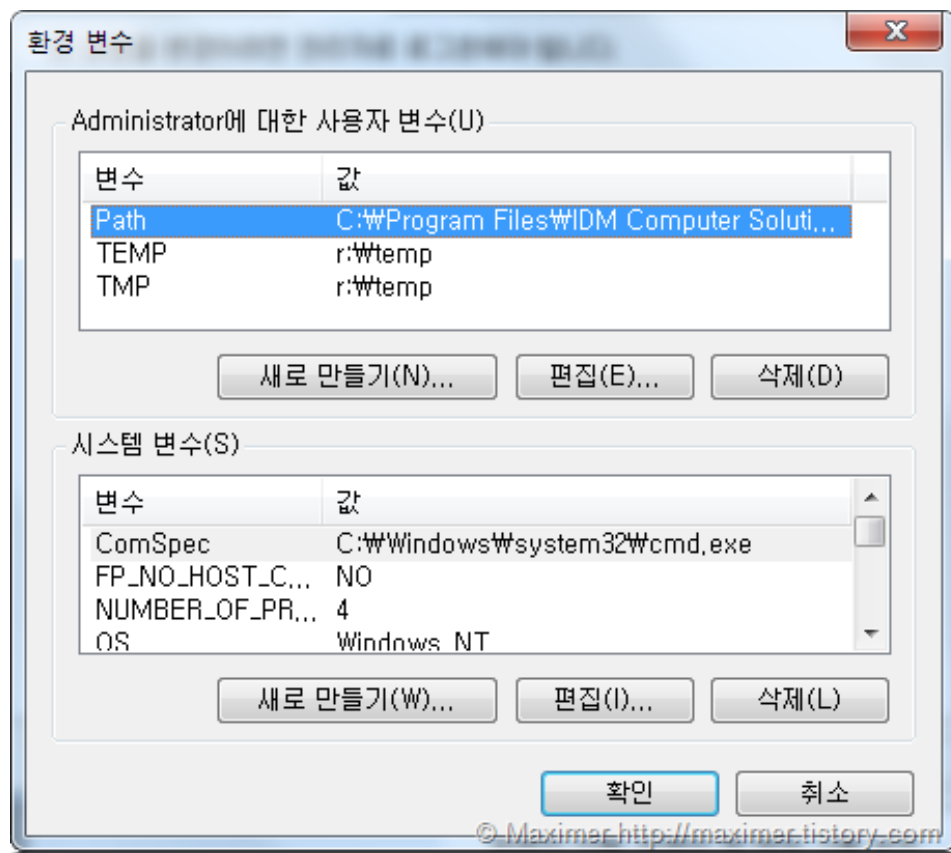
- Vi 에디터
 - 마우스 없이도 유연하게 파일 편집가능
 - 키보드 전체가 명령어(실제로 몇 개만 알아도 사용가능)
 - 대소문자에 따라 명령어 수행이 다름
- vi history
 - 유닉스가 나오고 에디터로 1976년 BY 빌조이
 - 기타 에디터로는 ed , pico , emacs
- vim ?(Vi Improved)
 - 리눅스에서는 전부 vi -> vim 으로 alias 되어있음
 - 일반적으로 유닉스에는 vim이 설치 안되어 있어 전통방식 vi 사용

윈도우 환경변수 설정

윈도우 환경변수

■ 환경변수

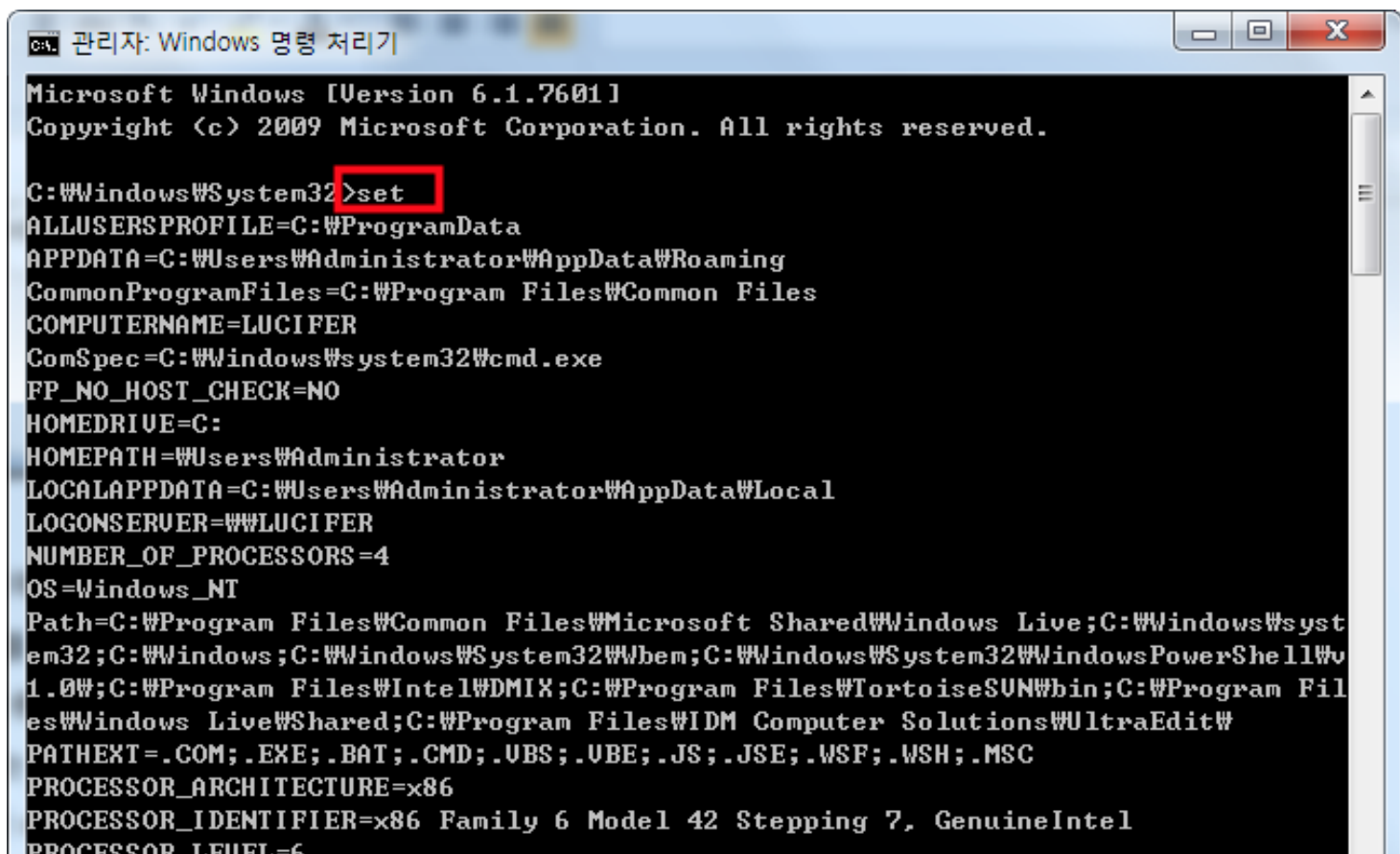
- 윈도우에는 특정경로나 실행파일에 대해 일정한 변수를 지니고 있는데, 이를 환경변수라고 한다. 아래처럼 유저가 추가 및 수정도 가능하다.



윈도우 환경변수

■ 환경변수

- 자신의 윈도우 환경변수를 알아내는 방법 → 커맨드창에서 **Set** 엔터



The screenshot shows a Windows Command Prompt window titled "관리자: Windows 명령 처리기". The command prompt displays the output of the 'set' command, which lists various system environment variables. The 'set' command is highlighted with a red box. The output includes variables like ALLUSERSPROFILE, APPDATA, CommonProgramFiles, COMPUTERNAME, ComSpec, FP_NO_HOST_CHECK, HOMEDRIVE, HOMEPATH, LOCALAPPDATA, LOGONSERVER, NUMBER_OF_PROCESSORS, OS, Path, PATHEXT, PROCESSOR_ARCHITECTURE, PROCESSOR_IDENTIFIER, and PROCESSOR_LEVEL.

```
C:\Windows\System32>set
Microsoft Windows [Version 6.1.7601]
Copyright (c) 2009 Microsoft Corporation. All rights reserved.

C:\Windows\System32>set
ALLUSERSPROFILE=C:\ProgramData
APPDATA=C:\Users\Administrator\AppData\Roaming
CommonProgramFiles=C:\Program Files\Common Files
COMPUTERNAME=LUCIFER
ComSpec=C:\Windows\system32\cmd.exe
FP_NO_HOST_CHECK=NO
HOMEDRIVE=C:
HOMEPATH=\Users\Administrator
LOCALAPPDATA=C:\Users\Administrator\AppData\Local
LOGONSERVER=\\LUCIFER
NUMBER_OF_PROCESSORS=4
OS=Windows_NT
Path=C:\Program Files\Common Files\Microsoft Shared\Windows Live;C:\Windows\system32;C:\Windows;C:\Windows\System32\Wbem;C:\Windows\System32\WindowsPowerShell\v1.0;C:\Program Files\Intel\DMIX;C:\Program Files\TortoiseSVN\bin;C:\Program Files\Windows Live\Shared;C:\Program Files\IDM Computer Solutions\UltraEdit\
PATHEXT=.COM;.EXE;.BAT;.CMD;.VBS;.VBE;.JS;.JSE;.WSF;.WSH;.MSC
PROCESSOR_ARCHITECTURE=x86
PROCESSOR_IDENTIFIER=x86 Family 6 Model 42 Stepping 7, GenuineIntel
PROCESSOR_LEVEL=6
```

윈도우 환경변수 - 윈도우7 환경변수

	값
ALLUSERSPROFILE	C:\ProgramData
APPDATA	C:\Users\Administrator\AppData\Roaming
CommonProgramFiles	C:\Program Files\Common Files
COMPUTERNAME	Maximer
ComSpec	C:\Windows\system32\cmd.exe
FP_NO_HOST_CHECK	NO
HOMEDRIVE	C:
HOMEPATH	\Users\Administrator
LOCALAPPDATA	C:\Users\Administrator\AppData\Local
LOGONSERVER	\\Maximer
NUMBER_OF_PROCESSORS	4
OS	Windows_NT
Path	C:\Program Files\Common Files\Microsoft Shared\Windows Live;C:\Windows\system32;C:\Windows;C:\Windows\System32\Wbem;C:\Windows\System32\WindowsPowerShell\Wv1.0\
PATHEXT	.COM;.EXE;.BAT;.CMD;.VBS;.VBE;.JS;.JSE;.WSF;.WSH;.MSC
PROCESSOR_ARCHITECTURE	x86
PROCESSOR_IDENTIFIER	x86 Family 6 Model 42 Stepping 7, GenuineIntel
PROCESSOR_LEVEL	6
PROCESSOR_REVISION	2a07
ProgramData	C:\ProgramData
ProgramFiles	C:\Program Files
PROMPT	\$P\$G
PSModulePath	C:\Windows\system32\WindowsPowerShell\Wv1.0\Modules\
PUBLIC	C:\Users\Public
SESSIONNAME	Console
SystemDrive	C:
SystemRoot	C:\Windows
TEMP	C:\Users\Administrator\AppData\Local\TEMP
TMP	C:\Users\Administrator\AppData\Local\TEMP
USERDOMAIN	Maximer
USERNAME	Administrator
USERPROFILE	C:\Users\Administrator
windir	C:\Windows
windows_tracing_flags	
windows_tracing_logfile	C:\WBVTBin\Tests\installpackage\csilogfile.log