Operating System

1장_운영체제의 개요

다양한 운영체제

1. 개인용 컴퓨터
마이크로소프트 윈도우, 애플 MacOS
(과거에는 마이크로소프트 MS-DOS, 애플 매킨토시)

2. 대형 컴퓨터
유닉스, 리눅스

3. 모바일 운영체제
(1) 애플의 ios
- 애플의 폐쇄 정책으로 애플 제품에만 사용
- 유닉스 기반

(2) 구글의 안드로이드
- 안드로이드의 개방 정책으로 여러 스마트폰 제조사에서 사용
- 유닉스 기반

4. 임베디드 운영체제(또는 임베디드 시스템)
MP3, 네비게이션, PMP

운영체제 소개

운영체제의 필요성

1. 운영체제가 없는 컴퓨터?

- 에니악
- 프로그래밍 가능했기 때문에 컴퓨터라고 부를 수 있었음
- 전선의 연결을 변경함으로써 프로그래밍(하드와이어링)
- 미사일 탄도를 계산

2. 운영체제가 있는 기계 VS 없는 기계 (스마트폰 VS 유선전화기)

(1) 없는 기계: 기능 추가 및 성능 향상이 불가

(2) 있는 기계: 기능 추가 및 성능 향상이 가능

3. 운영체제의 역할

(1) 컴퓨터 자원(하드웨어) 관리

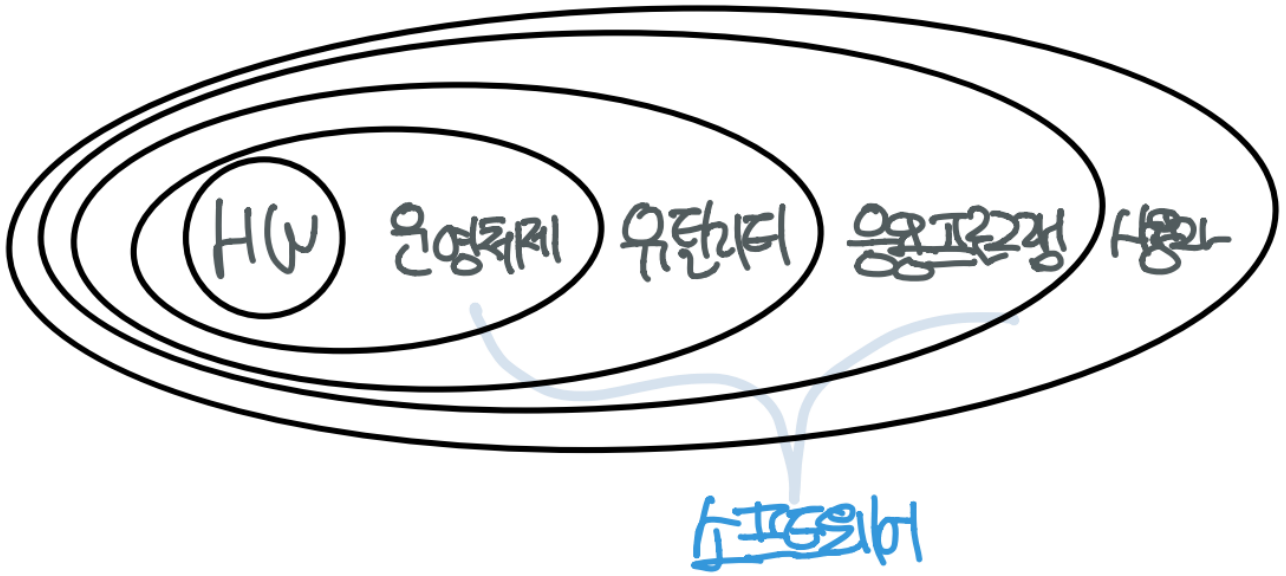
- 컴퓨터를 사용할 때 응용프로그램(워드, 웹브라우저 등) 동시 사용
- 하지만 컴퓨터를 구성하는 장치는 제한적
- 응용프로그램이 서로 독차지하려 함 => 운영체제가 중재자 역할

(2) 응용프로그램과 사용자에게 모든 자원을 숨김

운영체제 소개

(2) 응용프로그램과 사용자에게 모든 자원을 숨김
- 직접 접근하지 못하도록 하는 대신, 자원을 이용할 수 있는 인터페이스(커널의 시스템 콜) 제공

※ 펌웨어란?
- 운영체제는 SW
- 운영체제는 하드웨어를 조정하고 관리하는 역할
- 하드웨어를 조정하고 관리하려면 소프트웨어의 도움이 필요
- 운영체제를 펌웨어(하드웨어 + 소프트웨어)라고도 부름



운영체제 소개

운영체제의 역할 정리
1. 자원관리
- 효율성 추구

2. 자원보호
- 악성 프로그램으로부터 컴퓨터 자원과 다른 프로그램 보호하여 안정성 추구

3. 하드웨어 인터페이스 제공
- 확장성 추구
예) 무선 마우스 USB 꼽으면 별도의 SW 설치 없이 운영체제에서 제공
※ 하드웨어 드라이버 = 소프트웨어 드라이버 = 디바이스 드라이버 = 장치 제어기

4. 사용자 인터페이스
- 편리성 추구
- 과거의 유닉스, MS-DOS 운영체제는 키보드만 사용 가능
- 현재의 운영체제는 GUI제공하므로 마우스 및 스마트폰 터치 스크린 사용 가능

운영체제 소개

※ 유틸리티
- 악성 프로그램으로부터 컴퓨터 자원과 다른 프로그램 보호
- 시간이 지남에 따라 운영체제 기능과 복잡도 증가 및 악성 프로그램 침투 수법 발전
- 바이러스 검사, 디스크 조각 모음, 압축 프로그램과 같이, **운영체제 작업을 보조하는 소프트웨어**

운영체제 역사

1. 일괄 작업 시스템(batch job system 혹은 일괄 처리 시스템 batch process system)
- CPU, 메인 메모리 존재
- 키보드, 모니터, 입출력 장치 존재 X
- 천공 카드 리더기(입력 장치), 라인 프린터(출력 장치) 존재
- 모든 작업을 한꺼번에 처리
- 프로그램 중간에 데이터 입력/수정/출력 불가능
- 윈도우 운영체제가 부팅시 읽는 autoexec.bat에서 bat은 batch job을 의미
- autoexec.bat은 운영체제가 시작할 때 한번에 처리해야 할 작업

2. 대화형 시스템
- 키보드, 모니터, 입출력 장치 존재
- 프로그램 중간에 데이터 입력/수정/출력 가능

운영체제 역사

3. 시분할 시스템
- 시분할 시스템을 통해 다중 프로그래밍 기술, 다중 사용자 시스템 구현
- 다중 프로그래밍은 하나의 CPU로 여러 작업을 동시에 실행
- 단점으로는 중요한 작업이 일정 시간 안에 끝나는 것 보장 불가능

4. 분산 시스템
(1) 메인 프레임
고가의 컴퓨터

(2) 분산 시스템
- 값이 싸고 크기가 작은 컴퓨터들의 묶음
- 여러 컴퓨터로 작업을 처리하고 결과를 교환
- 대형 컴퓨터에 버금가는 시스템
- 단점으로는 모든 컴퓨터가 동일한 지위이기 때문에, 컴퓨터가 고장나거나 추가돼도 작업을 분배하기 어려움

운영체제 역사

5. 클라이언트/서버 시스템
- 분산 시스템 문제 해결
- 단점으로는 모든 요청이 서버로 집중되기 때문에 서버 과부하 문제점 존재

※ 데몬
- 멈추지 않고 계속 동작하는 프로그램
예) 클라이언트/서버 시스템에서 서버는 멈추지 않고 계속 동작

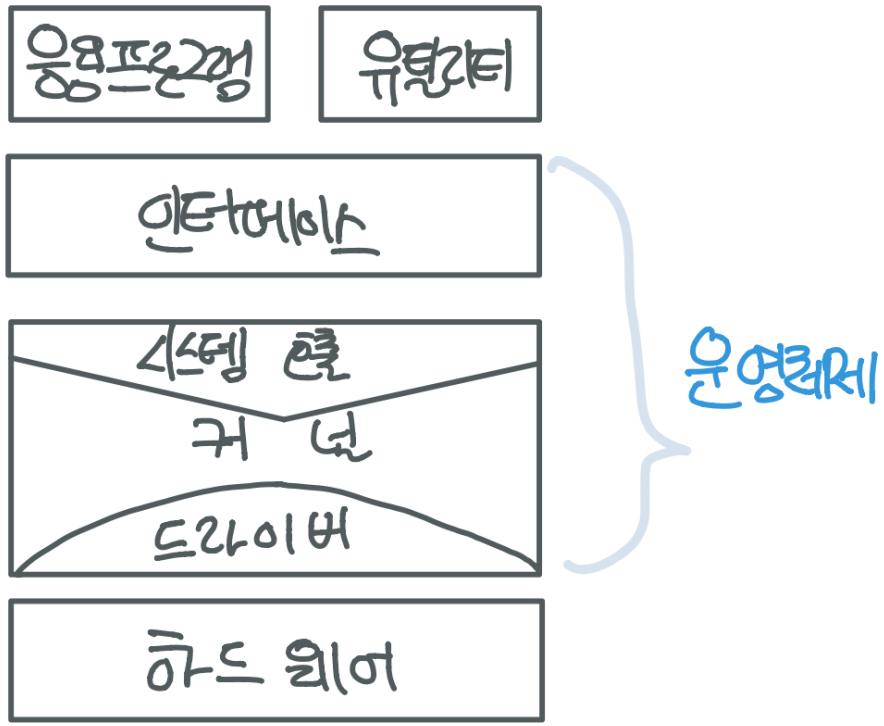
6. P2P시스템(Peer to Peer)
- 서버를 거치지 않고 사용자와 사용자를 직접 연결
- 사용자 간에 파일 전송이 이루어지기 때문에 서버의 부하가 적음

운영체제의 구조

1. 커널
- 운영체제의 핵심 기능
- 프로세스 관리/메모리 관리/입출력 관리 등,

2.인터페이스(GUI)
- 커널에 명령을 전달
- 사용자와 응용 프로그램에 실행 결과 전달
- 같은 커널을 사용하더라도 인터페이스가 다를 수 있음

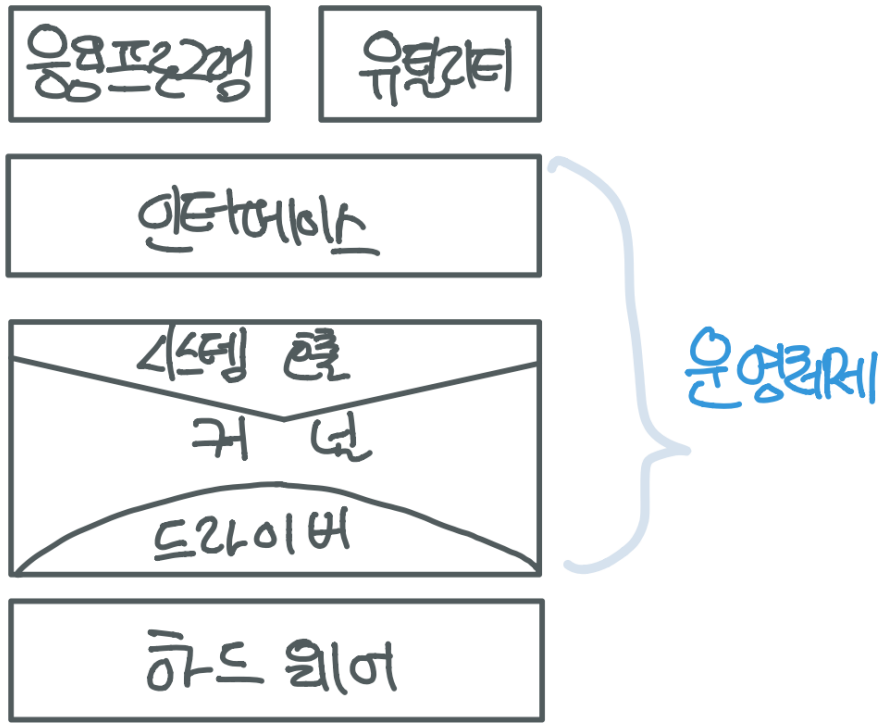
예) 유닉스 사용자 인터페이스는 '셸'로, 명령어 기반이라 사용하기 어려움
메킨토시 운영체제도 유닉스 커널 기반이지만 사용하기 쉬움



운영체제의 구조

커널 내부
(1) 시스템 콜
- 커널은 사용자나 응용 프로그램으로부터 컴퓨터 자원을 보호하기 위해 자원에 직접 접근하는 것을 차단
- 대신 시스템 자원의 사용과 연관된 함수 제공
예) 하드디스크에 데이터를 저장할 때 write()함수
데이터를 읽을 때 read()함수
- 사용자와 응용 프로그램은 데이터가 어디에 저장되는지 알 수 없음

(2) 디바이스 드라이버
하드웨어와 커널간 인터페이스



운영체제의 구조

※ 가상 머신
- C언어로 만든 운영체제는 유닉스
- C언어는 유닉스와 다른 커널을 가진 운영체제와의 호환성이 떨어짐
예) 유닉스에서 C언어로 작성된 코드는 윈도우에서 작동이 안됨
또한 유닉스 버전이 다양화됨에 따라 유닉스끼리도 호환이 안되는 경우 발생
- 호환성 문제를 해결한 언어가 자바
- 자바로 작성된 응용 프로그램은 운영체제 위의 가상머신 위에서 동작