

# 운영체제 개요

## - 운영체제 구조와 종류



컴퓨터소프트웨어학과  
김병국 교수

- 커널과 인터페이스를 안다.
- 시스템 호출과 디바이스 드라이버의 기능을 이해한다.
- 커널의 구조를 이해한다.
- 가상 머신을 안다.
- 운영체제의 종류와 특징을 안다.



# 목차

- 커널과 인터페이스
- 시스템 호출
- 디바이스 드라이버
- 커널의 구성
- 가상 머신
- 운영체제 종류



# 1. 커널과 인터페이스

## □ 커널(Kernel)

- 운영체제의 핵심(Core)
- 프로세스 관리, 메모리 관리, 저장장치 관리와 같은 운영체제의 필수 기능을 모아놓은 것



【컴퓨터 시스템의 구조】

## □ 인터페이스(IF: Interface)

- 커널에 사용자의 명령을 전달하고 실행 결과를 사용자에게 알려주는 역할
- GUI(Graphical User Interface) : 그래픽을 기반한 사용자 인터페이스



## 2. 시스템 호출 (1/4)

### □ 시스템 호출

- 커널이 자신을 보호하기 위해 만든 인터페이스
- 커널은 사용자나 응용 프로그램으로부터 컴퓨터 자원을 보호하기 위해 자원에 직접 접근하는 것을 차단



(a) 커피를 직접 만드는 경우

(b) 다른 사람에게 커피를 만들어 달라고 부탁하는 경우

**[커피주문을 통한 결과획득]**



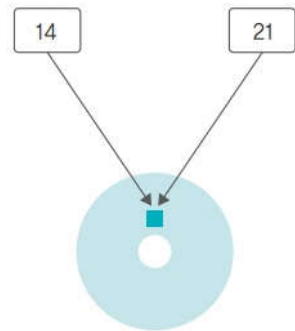
## 2. 시스템 호출 (2/4)

### □직접 접근

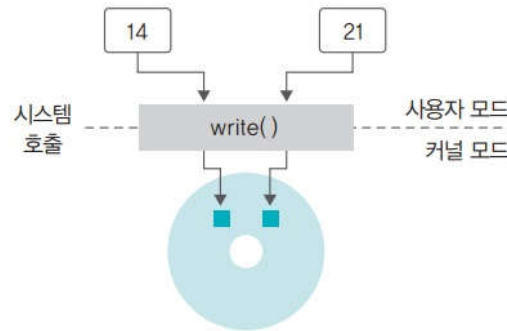
- 예1) 여러 응용 프로그램이 각자 지정한 위치에 데이터를 저장하려고 함
- 예2) 다른 응용 프로그램의 데이터가 변형되거나 의도치 않게 지워질 수도 있음

### □시스템 호출을 통한 접근

- 커널이 제공하는 write( ) 함수를 사용하여 데이터를 저장해달라고 요청
- 커널이 데이터를 가져오거나 저장하는 것을 전적으로 책임지기 때문에 컴퓨터 자원 관리가 수월



(a) 직접 접근



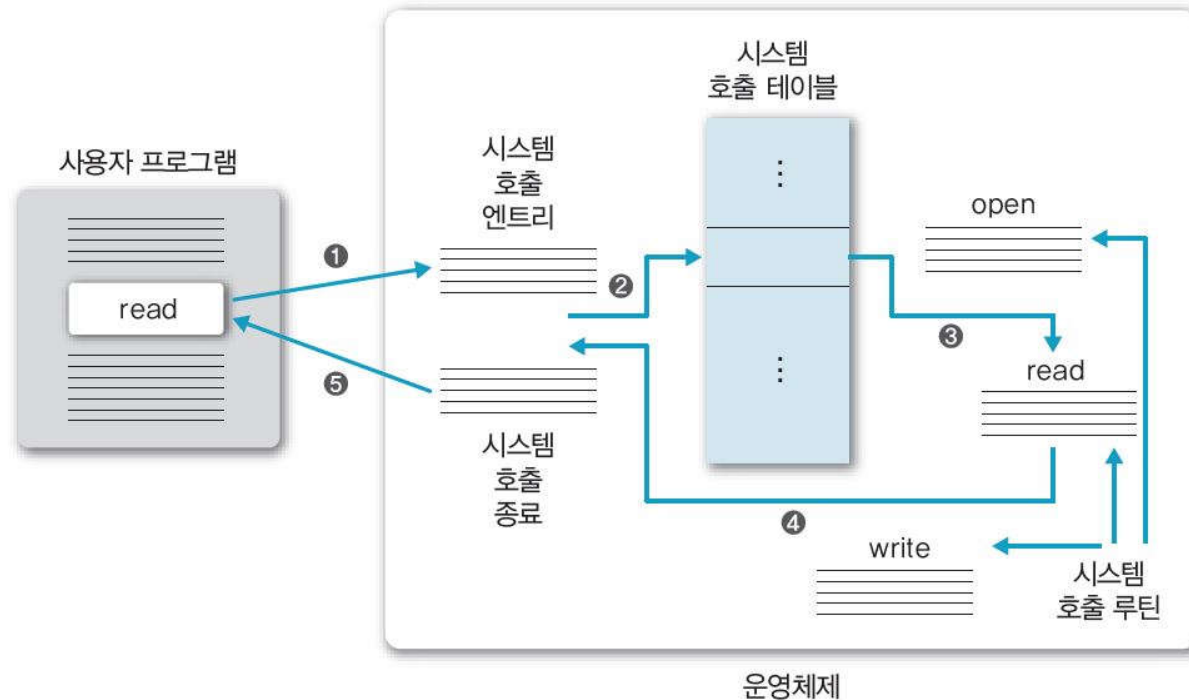
(b) 시스템 호출을 통한 접근

[접근 방식에 따른 동작구조]



## 2. 시스템 호출 (3/4)

### □ 시스템 호출 동작 방식



- ① 시스템 호출 서비스 요청
- ② 스위치 모드 : 서비스 인자와 검증
- ③ 시스템 호출 테이블을 이용하여 서비스 루틴으로 분기
- ④ 스위치 모드 : 서비스 루틴에서 반환
- ⑤ 시스템 호출에서 반환



## 2. 시스템 호출 (4/4)

### □ 정리

- 시스템 호출(System Call):
  - 커널(Kernel)이 제공하는 시스템 자원의 사용과 연관된 함수
  - 커널이 제공하는 인터페이스
  - 응용프로그램이 커널 영역에 진입할 수 있는 유일한 수단
- 실행 중인 프로그램과 운영체제 간의 인터페이스
  - 하드웨어 자원 접근 또는 운영체제가 서비스를 이용 할 때는 시스템 호출을 사용해야 함
- 운영체제는 커널 서비스를 시스템 호출로 제한
  - 다른 방법으로는 커널에 들어오지 못하도록 컴퓨터 자원을 보호
- 시스템 호출 방법
  - 프로그램에서 명령이나 서브루틴의 호출 형태로 호출
  - 시스템에서 명령 해석기를 사용하여 대화 형태로 호출
- 일반적인 시스템 호출
  - 프로세스 제어, 파일 조작, 장치 관리, 정보 유지 등





### 3. 디바이스 드라이버

#### □ 디바이스 드라이버(Device Driver)

- 커널과 하드웨어의 인터페이스 역할을 수행
- 마우스 같은 표준화된 간단한 제품은 커널이 기본 드라이버로 가지고 운영함
- 그래픽카드와 같이 복잡하고 제품에 따라 기능이 서로 다른 특징을 갖는 경우 제작사가 별도로 드라이버를 제공



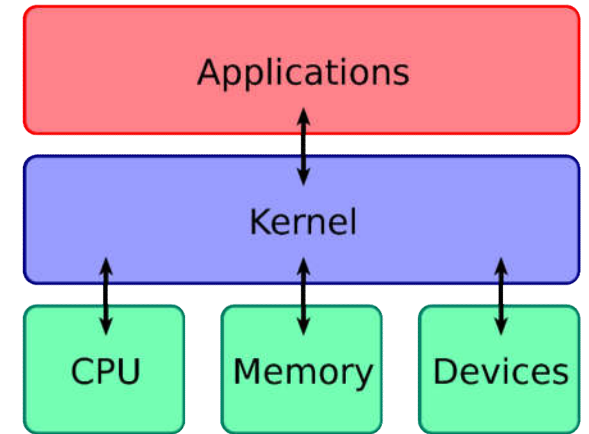
【운영체제의 구조】



## 4. 커널의 구조 (1/6)

### □ 커널의 핵심 기능

- 프로세스 관리
  - 프로세스에 CPU를 배분
  - 제반 환경을 제공
- 메모리 관리
  - 작업 공간을 배치
  - 가상의 공간을 제공
- 파일 시스템 관리
  - 데이터 저장을 위한 인터페이스를 제공
- 입출력 관리
  - 필요한 입력과 출력 서비스를 제공
- 프로세스간 통신(IPC) 관리
  - 공동 작업을 위한 각 프로세스간 통신 환경을 제공



## 4. 커널의 구조 (2/6)

### □특성

- 운영체제가 점점 더 다양한 하드웨어와 소프트웨어를 지원  
→ 커널 구조도 점점 복잡해 지고 있음
  - 복잡한 시스템은 설계, 구현, 테스트, 유지 보수 등 모든 면에서 어려움
- 이것의 해결을 위해 운영체제를 설계하는 다양한 방법 등장



## 4. 커널의 구조 (3/6)

### □ 단일형 구조 커널 (1/2)

- 초창기의 운영체제 구조
- 커널의 핵심 기능을 구현하는 모듈들이 구분 없이 하나로 구성



【단일형 구조 커널】



## 4. 커널의 구조 (4/6)

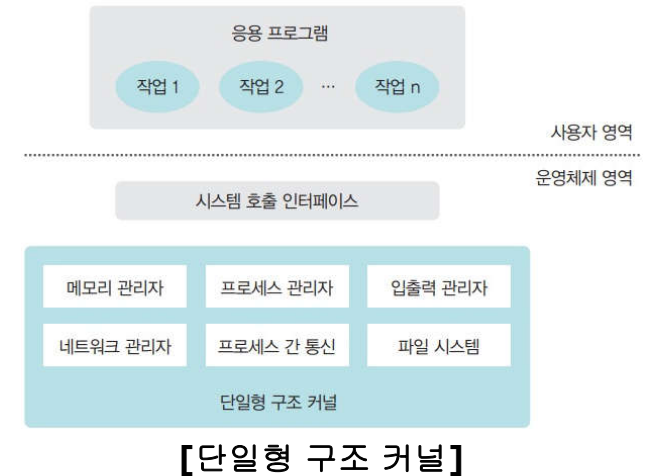
### □ 단일형 구조 커널 (2/2)

#### ■ 장점

- 모듈 간의 통신 비용이 줄어들어 효율적인 운영이 가능

#### ■ 단점

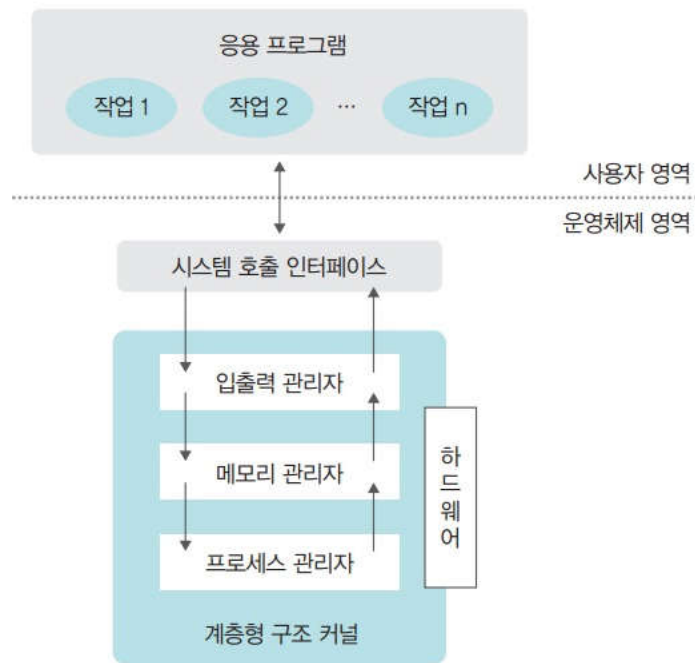
- 복잡한 구조로 인해 버그나 오류 수정이 어려움
- 운영체제의 여러 기능이 상호 의존성이 있으므로 기능상의 작은 결함이 시스템 전반에 미침
- 다양한 환경의 시스템에 적용하기 어려움 → 호환성 부족(Incompatibility)
- 현대의 운영체제 : 매우 크고 복잡하기 때문에 완전 단일형 구조의 운영체제를 구현하기가 어려움



## 4. 커널의 구조 (5/6)

### □ 계층형 구조 커널

- 유사 기능을 가진 모듈을 하나의 계층으로 구현
- 계층 간의 통신을 통해 운영체제 서비스를 지원



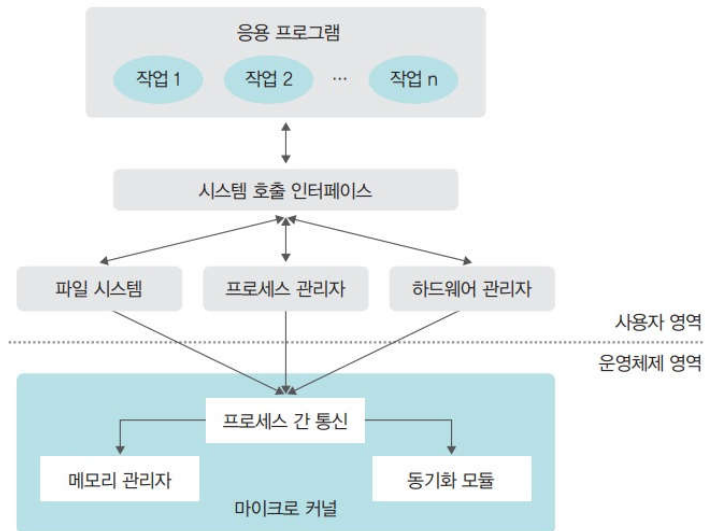
[계층형 구조 커널]



## 4. 커널의 구조 (6/6)

### □ 마이크로 구조 커널

- 가장 기본적인 기능만 제공
  - 기능 예: 프로세스 관리, 메모리 관리, 프로세스 간 통신 관리 등
- 커널의 각 모듈을 세분화
- 모듈 간의 정보 교환은 마치 프로세스 간 통신처럼 이루어짐



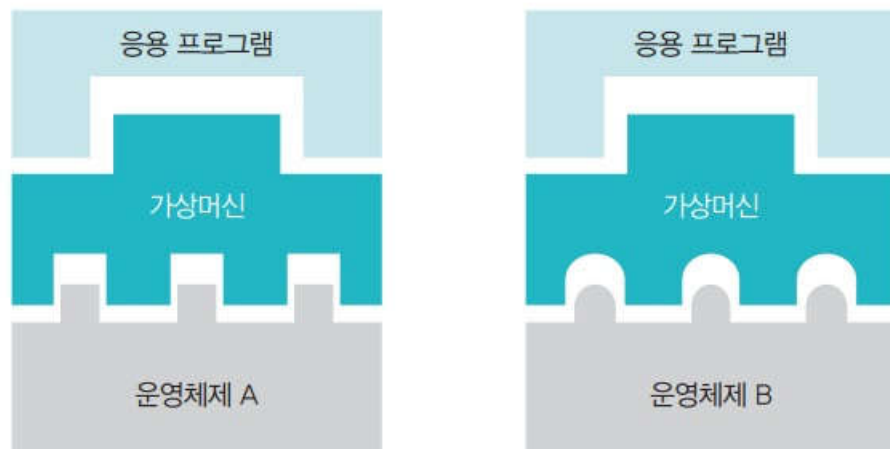
[마이크로 구조 커널]



# 5. 가상 머신

## □ 가상 머신(Virtual Machine)

- 가상 머신을 사용하면 응용 프로그램이 모두 동일한 환경에서 작동하는 것처럼 보임
- 운영체제와 응용 프로그램 사이에서 작동하는 중간 프로그램(미들웨어: Middleware)
  - 예1) 자바 가상 머신(JVM: Java Virtual Machine) : 플랫폼과 독립적으로 응용프로그램에 대한 구동이 가능
  - 예2) VMware, VirtualBox : 호스트 운영체제에서 타깃 운영체제의 동작을 가능케 함



【가상머신의 개념】

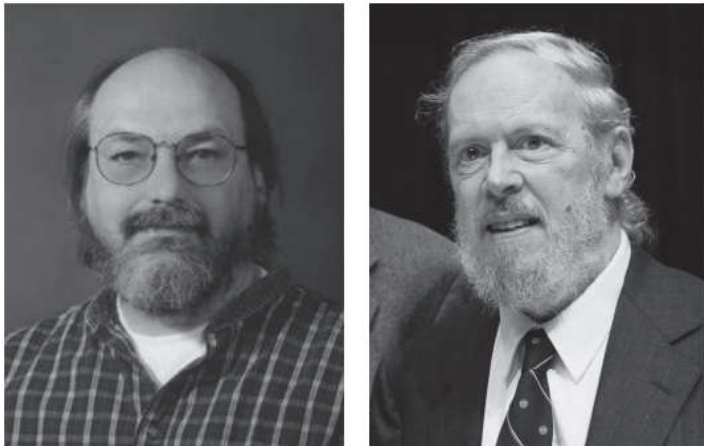




## 6. 운영체제의 종류(1/5)

### □ 유닉스의 개발과 확산

- 1969년 AT&T의 연구원으로서 멀틱스(Multics) 프로젝트의 결과물
- 켄 톰슨(Kenneth Lane Thompson)과 데니스 리치(Dennis MacAlistair Ritchie)에 의해 PDP-7 컴퓨터에 구현하려 함
- 멀틱스 프로젝트의 중간 산출물로 ‘**유닉스(UNIX)**’를 발표
- 유닉스의 뛰어난 기능으로 인해 현재까지 사용되고 있음
  - C언어 기반으로 개발됨으로써 쉬운 이식성으로 인해 다양한 컴퓨터에 이식이 될 수 있었음
  - 여러 기업과 대학에서 이를 이용한 연구가 진행되어 다양한 기능이 추가됨(대표: Solaris, FreeBSD)



[켄 톰슨과 데니스 리치]



## 6. 운영체제의 종류(2/5)

### □ 리눅스(Linux)의 개발

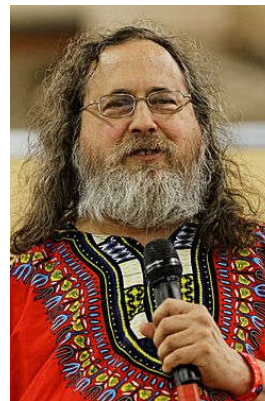
- 1991년에 리누스 툴바즈(Finland, University of Helsinki)에 의해 커널 개발
- PC(Intel 386)에서 동작하는 유닉스 호환 커널을 작성
  - 타넨바움의 미닉스(MINIX : Mini-UNIX)의 코드를 참고
- 리차드 스톨만의 GNU OS용 커널인 Hurd를 대체하기 위한 임시 커널로 채택
- 현재 GPL(GNU Public License)로 배포



[Andrew S. Tanenbaum]



[Linus B. Torvalds]



[Richard M. Stallman]



## 6. 운영체제의 종류(3/5)

### □ 애플 II의 등장

- 1976년 스티브 잡스는 애플(Apple Inc.)를 창업
- 1977년 애플 II라는 개인용 컴퓨터를 대중화 함
- 애플의 전략:
  - 기술을 발전시키는 것보다 사용자에게 편리함을 제공
  - GUI와 같은 사용자 편의를 위한 인터페이스 기술 개발에 집중



[애플 II]



## 6. 운영체제의 종류(4/5)

### □윈도우의 출시

- 마이크로소프트는 애플 Mac OS의 그래픽 사용자 인터페이스에 자극을 받아 윈도우 운영체제를 출시하고 계속 업그레이드하여 현재는 버전 10까지 나와 있음



【버전별 로고】



## 6. 운영체제의 종류(5/5)

### □ 안드로이드

- GNU의 리눅스를 사용하여 제작되었기 때문에 GPL을 따름
- 누구나 공짜로 사용할 수 있음
- 새로운 버전과 동시에 소스코드도 공개하기 때문에 누구나 수정 및 재배포가능
  - 안드로이드는 구글에서 개발됨
  - 스마트폰 제조사들은 대부분 안드로이드의 소스코드를 자사 제품에 맞게 수정하여 무료로 배포 중



[구글과 안드로이드]

※ 버전별 명칭:

1.0(Apple Pie, 2008.09), 1.1(Banana Bread, 2009.02), 1.5(Cupcake, 2009.04),  
1.6(Donut, 2009.09),  
2.0(Éclair, 2009.10), 2.2(Froyo, 2010.05), 2.3(GingerBread, 2010.10),  
3.0(Honeycomb, 2011.02),  
4.0(Ice Cream Sandwich, 2011.10), 4.1(Jelly Bean, 2012.06), 4.4(KitKat, 2013.10),  
5.0(Lollipop, 2014.10), 6.0(Marshmallow, 2015.10), 7.0(Nougat, 2016.08), 8.0(Oreo, 2017.08), 9(Pie, 2018.08), 10(Queen Cake, 2019.09), 11(Red Velvet Cake, 2020.02)

12(Snow Cone, 2021.10.03)

13(Tiramisu, 2022.3Q)





**수고하셨습니다.**

