Ch01_운영체제 소개

운영체제론

학습목표

- ✓ 01_ 소개
- ✓ 02_운영체제란 무엇인가
- ✓ 03_초기 역사 : 1940, 1950년대
- ✓ 04_1960년대
- ✓ 05_1970년대
- ✓ 06_1980년대
- ✓ 07_인터넷과 월드 와이드 웹의 역사
- ✓ 08_1990년대
- ✓ 09_2000년 이후
- ✓ 10_응용 프로그램 기반
- ✓ 11_운영체제 환경
- ✓ 12_운영체제의 구성 요소와 목표
- ✓ 13_운영체제 아키텍처

01_소개

- 수십 년 동안 빠르게 진화해온 컴퓨터 분야
- 사용자 워크스테이션 BIPS(Billions of Instructions per second)의 처리 속도
- 일상 생활의 모든 영역에 컴퓨터 사용
  - 운영체제의 역할과 책임 변화
    - ✓문서, 게임, 음악, 비디오, 자산 관리
    - ✓노트북, PDA, 휴대폰, MP3
- 이 책에서 다룰 내용
  - 운영 체제의 원리
  - 운영 체제의 구조와 책임
  - 운영 체제 설계 시 고려사항
  - 분산 컴퓨팅과 관련하여 운영 체제 설계 이슈

02_운영체제란 무엇인가

- 1960년대 운영 체제는 하드웨어를 제어하기 위한 소프트웨어

- 응용 프로그램이 하드웨어와 상호 작용할 수 있는 해주는 소프트웨어
  - 주어진 입력에 맞는 결과를 보장하도록 소프트웨어와 하드웨어 조작

- 운영체제는 기본적으로 자원 관리자
  - 하드웨어
    - ✓ 프로세서
    - ✓ 메모리
    - ✓ 입출력 장치
    - ✓ 통신 장치
  - 소프트웨어

03_초기 역사 : 1940, 1950년대

□ 뚜렷한 변화를 보이며 발전한 운영체제

- 1940년대
  - 초기 디지털 컴퓨터는 운영체제를 포함하지 않음

- 1950년대
  - 한번에 한가지 작업만 수행
  - 사용 효율을 향상시키기 위해 작업 사이의 이동을 원활하게 하는 기술 사용
  - 단일 스트림 배치 처리 시스템(single-stream batch-processing system)
  - 시스템 자원을 직접 제어

04_1960년대

□ 배치 처리 시스템

□ 멀티 프로그래밍

- 자원 공유
  - 운영 체제는 여러 작업을 동시에 처리하는 방향으로 발전

□ 1964년 IBM System/360 발표

□ 대화식 사용자(interactive user)

□ 시분할 시스템

- 대화식 컴퓨팅 지원, 프로그램과 데이터 공유
  - MIT의 CTSS(compatible Time-sharing System)
  - IBM의 TSS(time sharing System)
  - CTSS, CP/CMS(control program/conversational System)
  - 멀틱스(multics)

05_1970년대

□ 멀티 모드 멀티프로그래밍 시스템

- 배치 처리와 시분할, 실시간 응용 프로그램 지원
- 1970년 후반에는 개인 컴퓨터 혁명 시작

□ TCP/IP 통신 표준 활성화

- 이더넷의 등장으로 근거리 통신망 환경 발전
- 보안 문제 등장
  - 암호 작성/해독 기술 주목

□ 운영체제가 네트워크와 보안을 아우르는 수준으로 발전

- 상업적 요구 충족할 만큼 개선

06_1980년대

□ 개인용 컴퓨터와 워크스테이션의 시대

- 소형 컴퓨터 중앙 처리 장치인 마이크로프로세스 기술 발전
- 소프트웨어의 발전
  - 스프레드시트, 워드 프로세서, 데이터베이스, 그래픽 패키지
- 그래픽 사용자 인터페이스(graphical user interface) 등장

□ 네트워크 기술의 발전

- 컴퓨터간 정보 전송 경제적이고 현실적 수준의 전환
- 클라이언트/서버 컴퓨팅 모델의 널리 보급
  - 분산 컴퓨팅 환경 제공

□ 소프트웨어 공학 분야의 발전

07_인터넷과 월드 와이드 웹의 역사

ARPA(Advanced Research Project Agency)

- **ARPAnet 구현**
  - 오늘날 인터넷의 시조
  - 빠르고 쉬운 통신 능력
  - 중앙 집중적인 통제 없이 작동
    - ✓네트워크의 일부가 고장나도 남은 부분은 다른 경로로 전송
  - TCP/IP 프로토콜
    - ✓오류 없는 전송 보장
  - 상업용 목적으로 활용
    - ✓대역폭 증가
    - ✓하드웨어와 통신비용 감소
- **World Wide Web(WWW)**
  - 사용자는 모든 주제에 대해 멀티미디어 기반 문서 조회
  - 1989년 CERN에서 하이퍼링크 기반 문서 공유 방법 개발 시작

08_1990년대

1990년 대 특성

- 하드웨어 성능의 기하급수적 발전
  - 프로세싱 파워과 저장 공간의 증가
- 월드 와이드 웹의 탄생으로 분산 컴퓨팅의 증가
  - 개인 컴퓨터간에도 분산 컴퓨팅이 일반적인 일
  - 자원의 사용 확대 및 효율 향상
  - 네트워크 속도가 컴퓨터의 처리 속도에 미치지 못한 한계
- Microsoft의 성장
  - Windows 운영체제
  - 사용자 친화적 인터페이스 제공
  - 데스크탑 운영체제 잠식
  - 기업용 운영체제 Window NT 출시
- 객체 기술
  - 객체 지향 프로그램의 등장은 컴포넌트의 재사용 가능하게 하여 개발 시간 단축
  - 객체 지향 운영체제의 등장으로 기존 운영체제보다 유지보수와 확장이 용이
- 오픈 소스 운동
  - 오픈 소스 소프트웨어를 지향하는 운동
    - ✓ 리눅스, 아파치 등

09_2000년 이후

□ 미들웨어

- 네트워크를 통해 독립적인 응용 프로그램을 서로 연결하는 소프트웨어

□ 웹 서비스

- 분산 컴퓨팅으로 전환 촉진

□ 멀티프로세스와 네트워크 아키텍처

- 새로운 하드웨어와 소프트웨어 설계 기술 개발 기회 제공

□ 고도 병렬성(massive parallelism)

- 프로세서를 다수 보유해 여러 독립적인 계산으로 병렬로 수행

□ 운영체제 인터페이스 표준화

- 다양한 프로그램 지원하고 사용 용이
  - Windows xp 운영체제의 전문가와 사용자 집단 흡수
  - MS 차세대 운영체제 롱혼(longhorn)에서는 다른 유형 파일 포맷 통합 계획

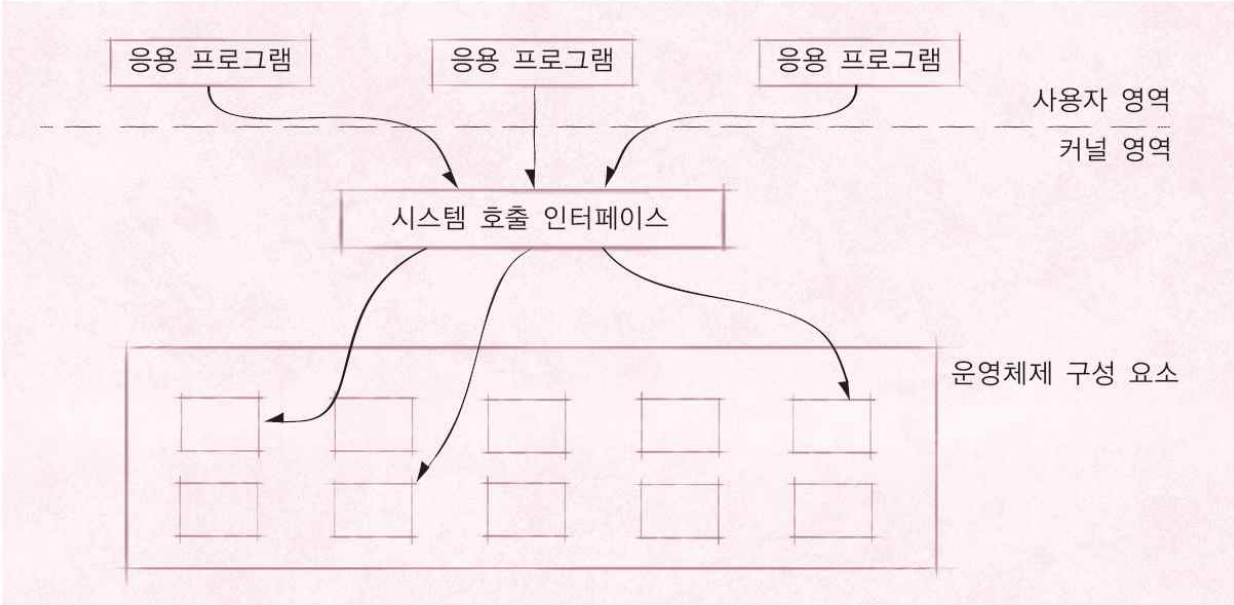
10_응용 프로그램 기반

IBM 개인용 컴퓨터는 거대한 소프트웨어 산업을 활성화

- ISV(independent software vendors) DOS에서 동작하는 IBM PC용 응용 발표

응용 프로그램 기반

- 운영체제는 응용 소프트웨어 개발자들의 메모리관리 입출력 회선 관리 부담 해소
  - API 제공으로 하드웨어 조작을 간단히 해결
- 응용 프



[그림 1-1] 응용 프로그램과 운영체제의 상호 작용

11_운영체제 환경

□ 범용 컴퓨터

- 대용량 메모리와 디스크, 고속 프로세서, 주변 장치로 구성
- 개인용 컴퓨터나 웨크스테이션으로 사용
- 고성능 하드웨어를 갖춘 고사양 웹 서버와 데이터베이스 서버에 적용 가능
- 대용량 메모리와 특수 목적 하드웨어, 여러 프로세스를 지원

□ 임베디드 시스템

- 휴대폰, PDA 같은 소형 기기에 기능을 제공하는 특화된 소규모 자원
- 효율적인 자원 관리 요구
- 적은 코드로 서비스 제공
- 전력관리
- 사용자 친화적 인터페이스 제공

□ 실시간 시스템

- 정해진 시간 안에 특정 작업 완료
  - 소프트 실시간 시스템
  - 하드 실시간 시스템

11_운영체제 환경

☐ 가상 머신

- 컴퓨터를 소프트웨어로 추상화

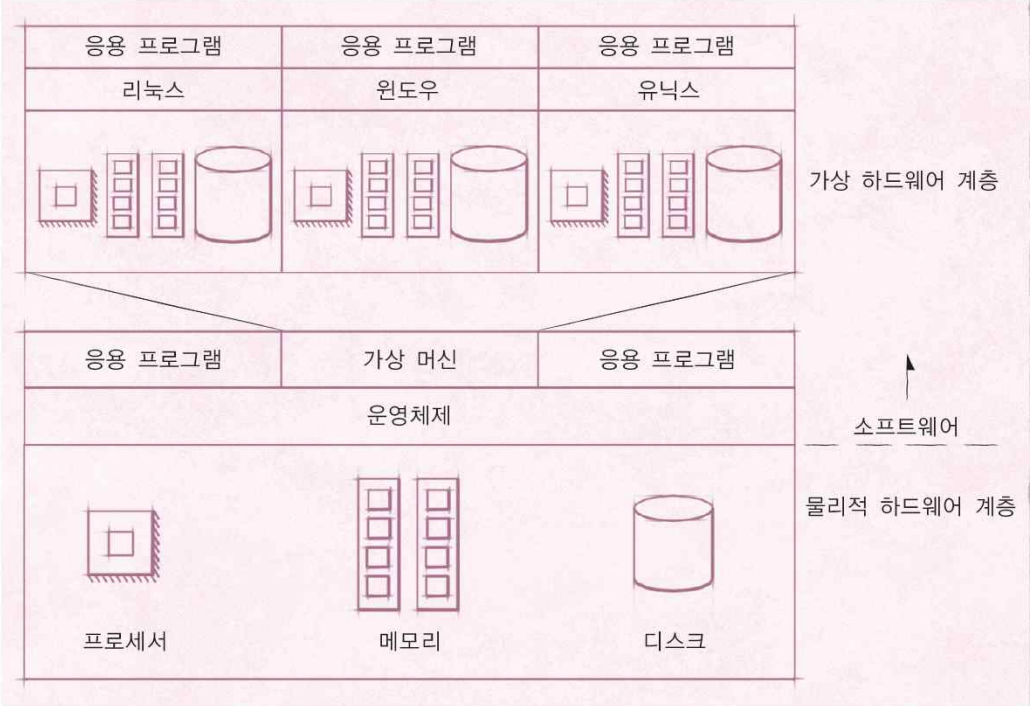
- 가상 머신 운영 체제
  - 가상 머신에 의해서 제공되는 자원관리

- 가상 머신의 응용 프로그램
  - 여러 운영체제의 인스턴스를 동시에 실행 가능하도록 허용
  - 에뮬레이션
    - ✓시스템에 존재하지 않는 하드웨어나 소프트웨어 가능 흉내

- 여러 사용자가 하드웨어 공유
  - 소프트웨어 이식성 향상

- 자바 가상 머신
  - JVM
  - VMware

11_운영체제 환경



[그림 1-2] 가상 머신 개념도

12_운영체제의 구성 요소와 목표

□ 운영체제의 핵심 구성 요소

- 프로세스 스케줄러
- 메모리 관리자
- 입출력 관리자
- 프로세스 간 통신 관리자
- 파일 시스템 관리자

□ 운영체제의 목표

- 효율성(efficiency)
- 견고함(robustness)
- 규모 확장성(scalability)
- 확장성(extensibility)
- 이식성(portability)
- 보안(security)
- 상호 작용성(interactivity)
- 사용성(usability)

13_운영체제 아키텍처

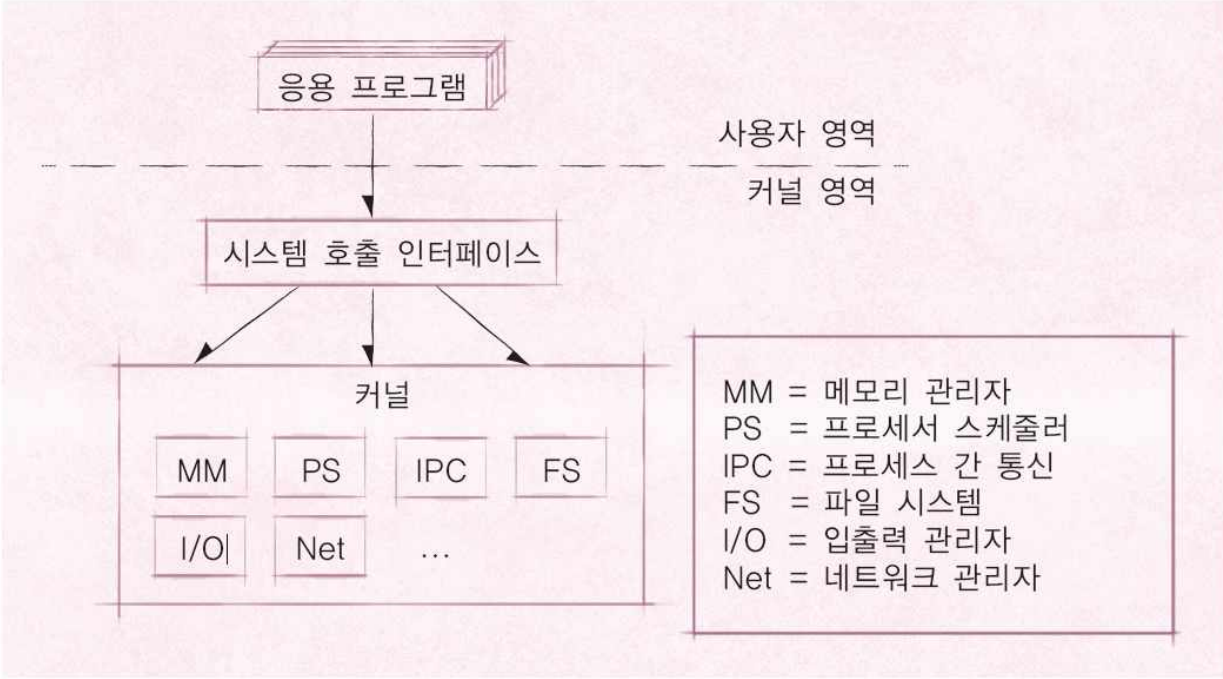
□ 현대 운영체제는 복잡

- 다양한 서비스 제공
- 다양한 하드웨어와 소프트웨어 지원
- 운영체제의 구성 요소의 실행 권한을 지정하여 복잡성 해결
  - 모놀리식 커널
  - 마이크로커널

13_운영체제 아키텍처

□ 모놀리식 아키텍처

- 모든 컴포넌트 커널에 포함
- 호출 기능만으로 다른 구성 요소와 통신 가능
- 컴퓨터 시스템에 제한 없이 접근
- 높은 성능
- 오류나 악성 코드에 취약

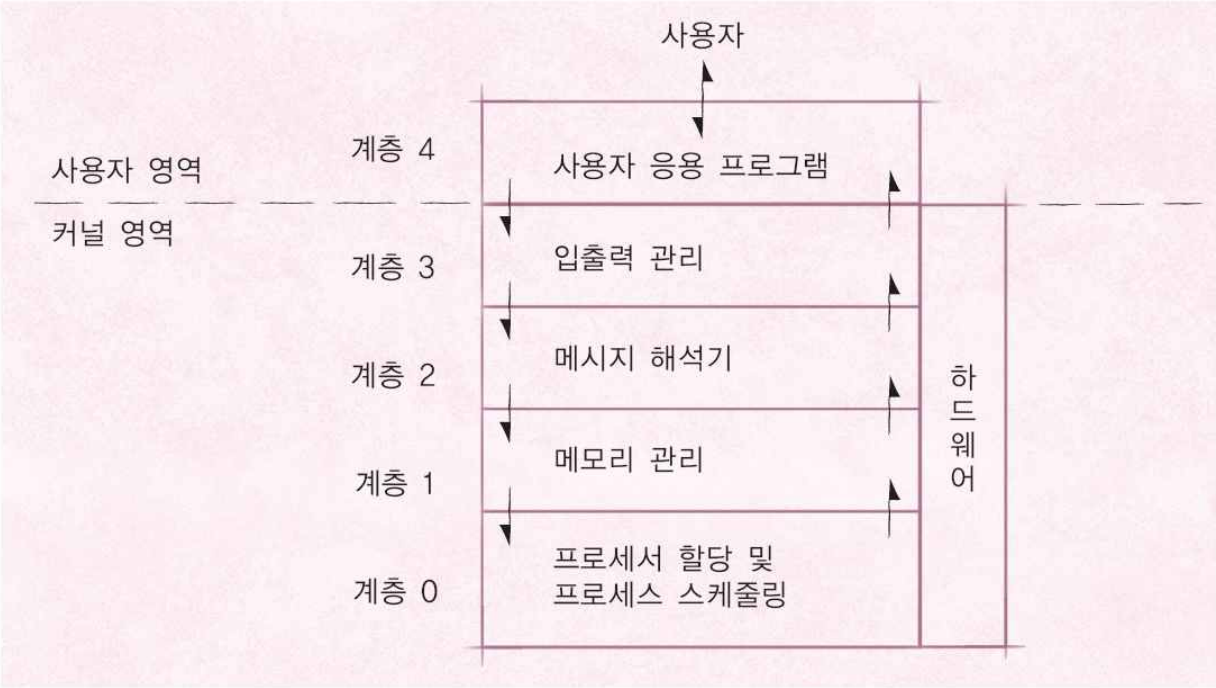


[그림 1-3] 모놀리식 운영체제 커널의 아키텍처

13_운영체제 아키텍처

□ 계층적 아키텍처

- 유사한 기능을 수행하는 요소 그룹으로 묶어 계층 구분
- 각 계층은 바로 상위 또는 하위 계층과 상호 작용
- 하위계층은 구체적인 구현 숨기고 인터페이스만 제공
- 운영체제의 구조화와 일관성 부여
- 소프트웨어 검증과 디버깅 및 수정 과정 간편



[그림 1-4] THE 운영체제의 계층

13_운영체제 아키텍처

□ 마이크로커널 아키텍처

- 소수의 서비스만 제공
  - 커널 규모 감소, 규모 확장성 향상

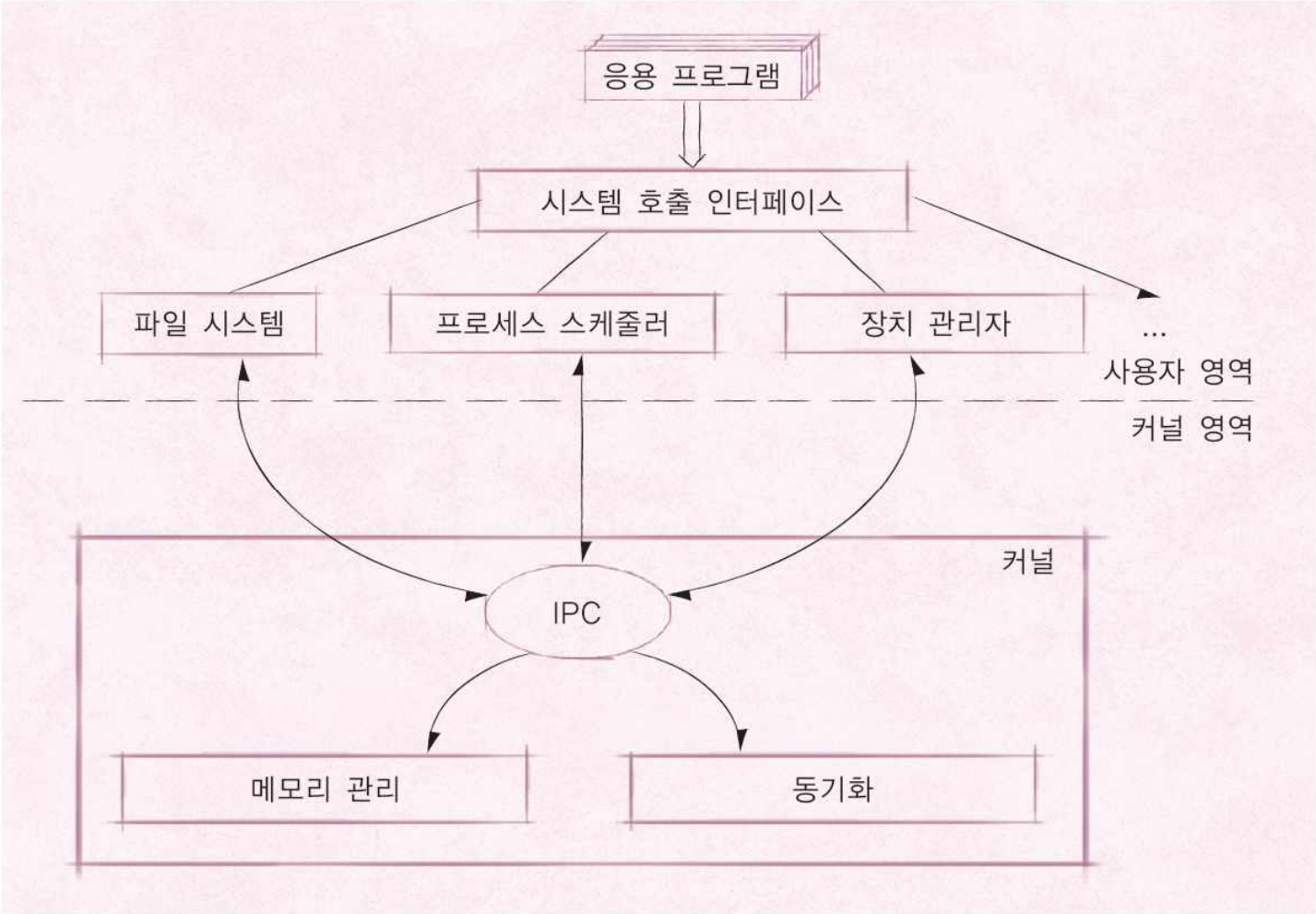
- 저수준 메모리 관리, 프로세스 간 통신, 프로세스 간 협력을 위한 동기화 기능
- 구성 요소를 낮은 수준의 권한으로 커널 외부에서 실행
- 확장성, 이식성, 규모 확장성 향상
- 모듈 간의 통신이 많아 성능 감소 우려

□ 네트워크 운영체제와 분산 운영체제

- 네트워크 운영체제
  - 네트워크에 있는 다른 컴퓨터의 자원에 접근 가능
  - 네트워크 파일 시스템

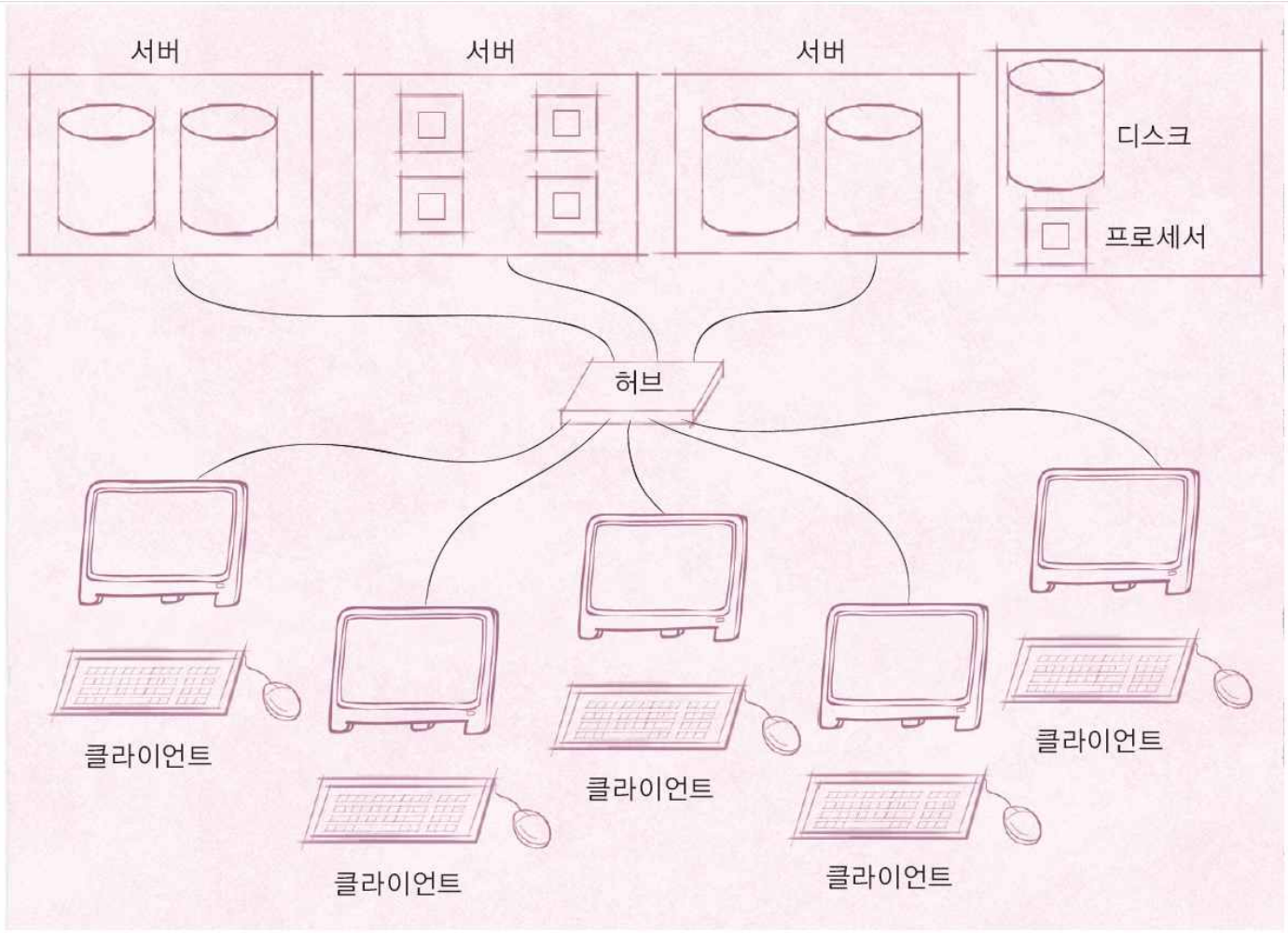
- 분산 운영체제
  - 한 대 이상의 컴퓨터에 있는 자원을 관리하는 특별한 운영체제
  - 구현 복잡
  - 프로세스가 공유 데이터에 접근하기 때문에 복잡한 알고리즘 요구

13_운영체제 아키텍처



[그림 1-5] 마이크로커널 운영체제 아키텍처

13_운영체제 아키텍처



[그림 1-6] 클라이언트/서버 기반 네트워크 운영체제 모델

Thank You !

운영체제론