리눅스 프로그래밍

운영체제 개요

인공지능소프트웨어과
이혜정 교수

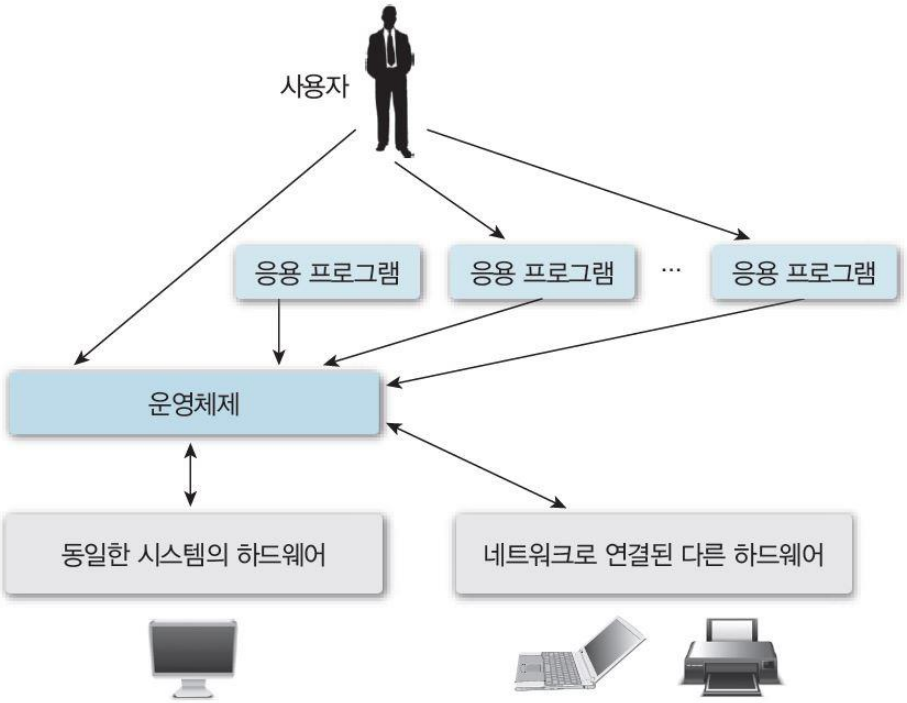
* 본 교안의 일부 그림과 텍스트는 '운영체제 – 그림으로 배우는 구조와 원리' (개정3판, 한빛미디어)으로 발췌하였음

목차

운영체제 소개

운영체제의 정의

- **사용자/응용 프로그램과 하드웨어 사이의 중간 매개체로 동작하는 시스템 소프트웨어**
  - 사용자가 응용 프로그램을 실행할 수 있는 기반 환경을 제공하여 컴퓨터를 편리하게 사용할 수 있도록 도와주고, 하드웨어를 효율적으로 사용할 수 있도록 다양한 기능을 제공하는 소프트웨어
  - 사용자가 하드웨어에 접근할 수 있는 유일한 수단



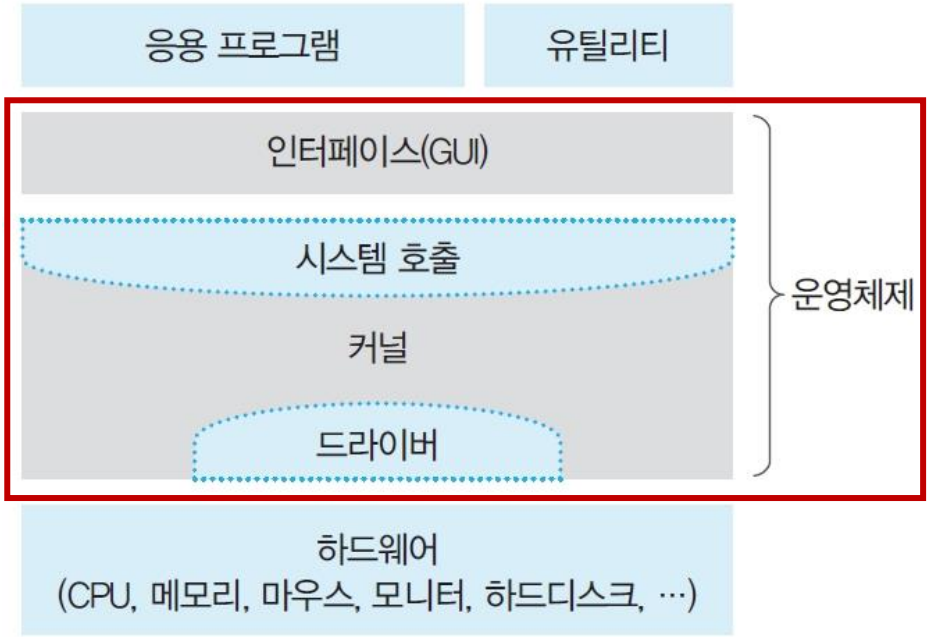
운영체제의 기능

- **응용 프로그램의 실행을 제어하고, 자원을 할당 및 관리하며, 입출력 제어 및 데이터 관리와 같은 서비스를 제공하는 소프트웨어**
  - 사용편리성을 위한 인터페이스 제공
    - 하드웨어 및 사용자, 응용 프로그램, 시스템 프로그램 사이에서 인터페이스
  - 효과적/효율적 자원 관리
    - 프로세서, 메모리, 입출력장치, 통신장치 등
  - 응용프로그램 실행 제어
    - 메일 전송, 파일 시스템 검사, 서버 작업 등
  - 시스템 보호
    - 다양한 사용자에게서 컴퓨터 시스템을 보호하려고 입출력을 제어하며 데이터를 관리

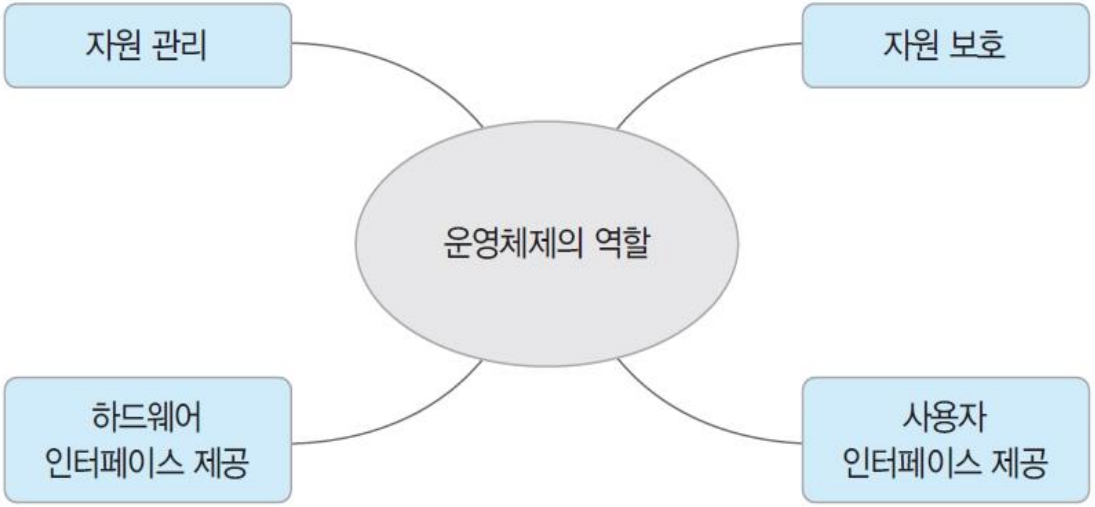
운영체제의 구성요소

- **커널**
  - 프로세스 관리, 메모리 관리, 저장장치 관리와 같은 운영체제의 핵심적인 기능을 모아놓은 것
  - 시스템 호출 (System Call)
    - 커널이 자신을 보호하기 위해 만든 인터페이스
    - 커널은 사용자나 응용 프로그램으로부터 컴퓨터 자원을 보호하기 위해 자원에 직접 접근하는 것을 차단
  - 디바이스 드라이버 (Device Driver)
    - 커널과 하드웨어의 인터페이스를 담당하며 드라이버라고도 불림
    - 마우스와 같이 간단한 제품은 드라이버를 커널이 가지고 있으나, 그래픽카드와 같이 복잡한 하드웨어의 경우 제작자가 드라이버를 제공함

- **인터페이스**
  - 커널에 사용자의 명령을 전달하고 실행 결과를 사용자에게 알려주는 역할
  - GUI (Graphic User Interface)
  - CLI (Command Line Interface)



운영체제의 역할



- **자원 관리**
  - 컴퓨터 시스템 자원을 응용 프로그램에 나눠 주어 사용자가 원활히 작업하게 함
  - 자원 요청한 프로그램이 여러 개면 적당한 순서로 자원 배분하고 적절한 시점에 자원 회수하여 다른 응용 프로그램에 나눠 줌
- **자원 보호**
  - 비정상적 작업으로부터 컴퓨터 자원 보호
- **하드웨어 인터페이스 제공**
  - 사용자가 복잡한 과정 없이 다양한 장치를 사용할 수 있게 하드웨어 인터페이스 제공
  - CPU, 메모리, 키보드, 마우스 같은 다양한 하드웨어를 일관된 방법으로 사용하도록 지원
- **사용자 인터페이스 제공**
  - 사용자가 운영체제를 편리하게 사용하도록 지원 예) 윈도우의 그래픽 사용자 인터페이스(GUI)

운영체제의 목표

- **효율성**
  - 같은 자원으로 더 많은 작업량 처리하거나, 같은 작업량 처리하는 데 더 적은 자원 사용
- **안정성**
  - 사용자와 응용 프로그램의 안전 문제와 하드웨어적인 보안 문제 처리
  - 시스템에 문제 발생시 이전으로 복구하는 결함 포용 기능 수행
- **확장성**
  - 다양한 시스템 자원을 컴퓨터에 추가/제거하기 편리한 것
- **편리성**
  - 사용자가 편리하게 작업할 수 있는 환경 제공하는 것

운영체제의 발전 목적

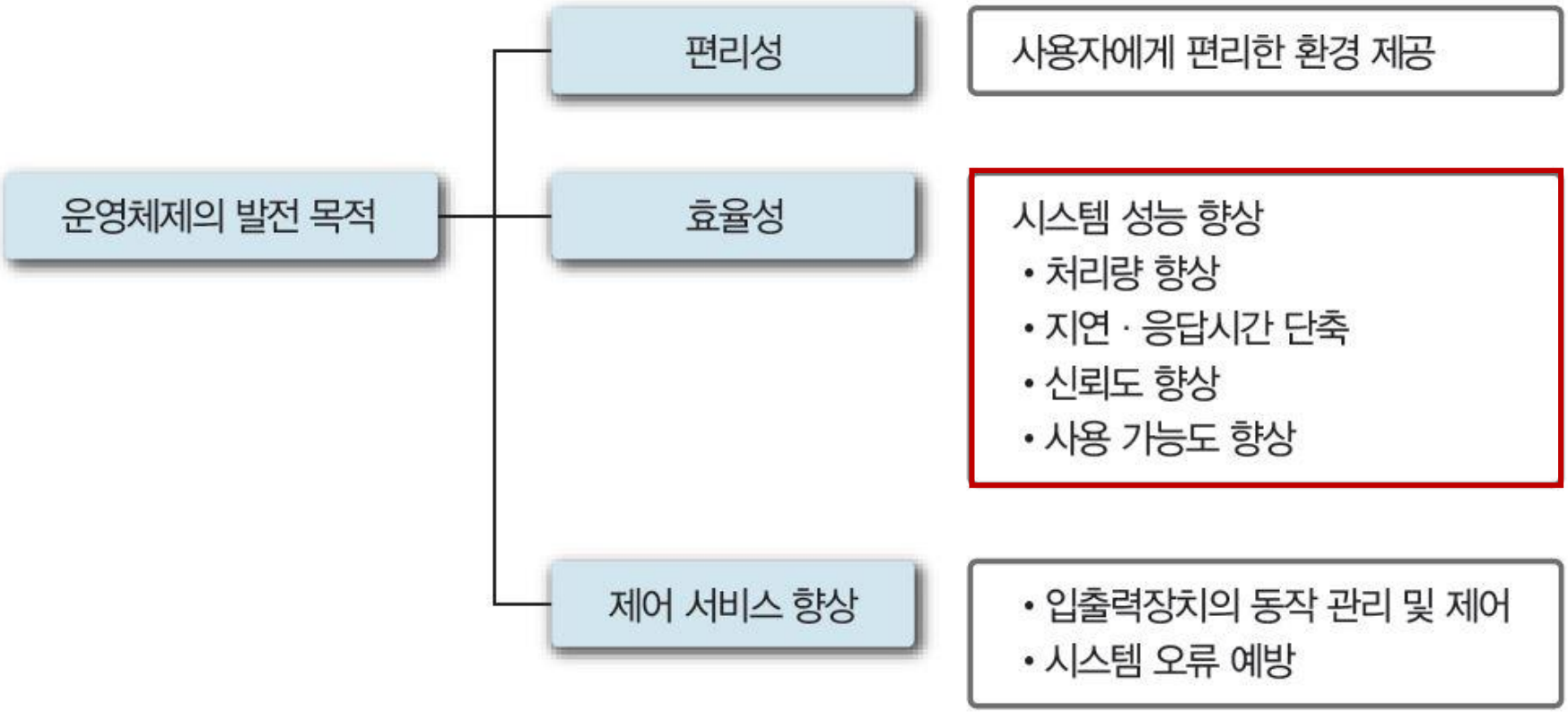
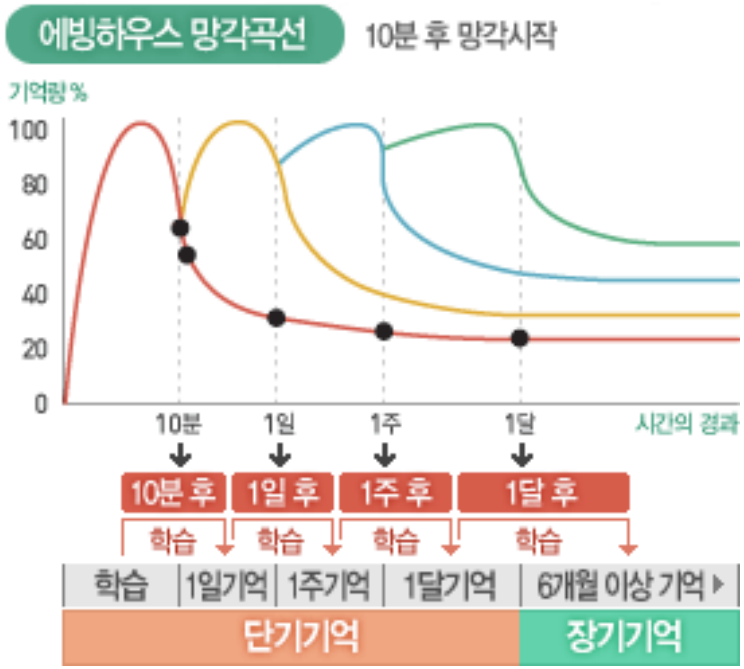


그림 2-3 운영체제의 발전 목적

망각 곡선 되살리기


에빙하우스 망각곡선
10분 후 망각시작
기억량 %
100
80
60
40
20
0
10분
1일
1주
1달
시간의 경과
10분 후
1일 후
1주 후
1달 후
학습
학습
학습
학습
학습
1일기억
1주기억
1달기억
6개월 이상 기억
단기기억
장기기억

운영체제 운용 기법

운영체제 유형 (운용 기법)

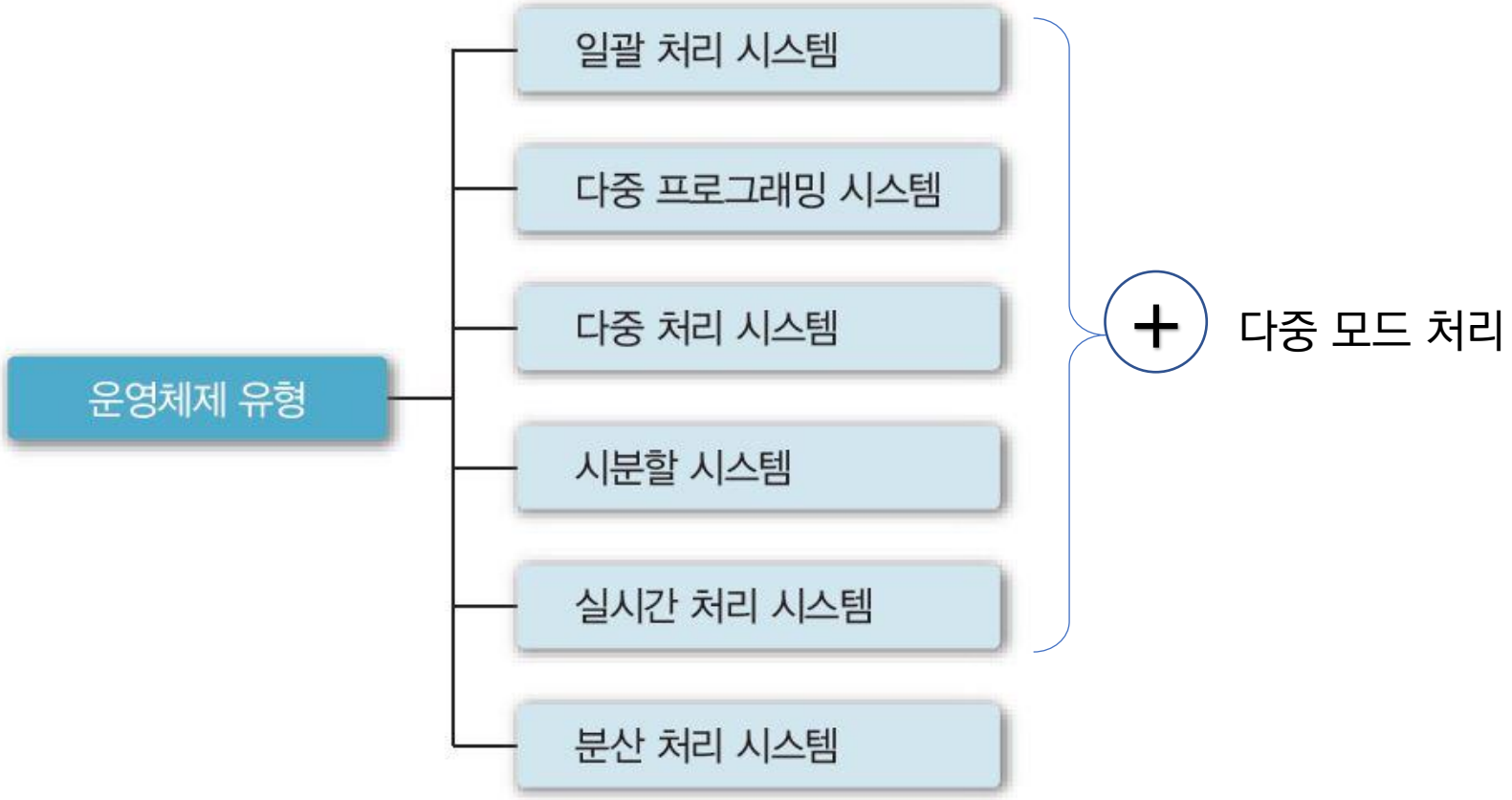


그림 2-11 운영체제의 유형

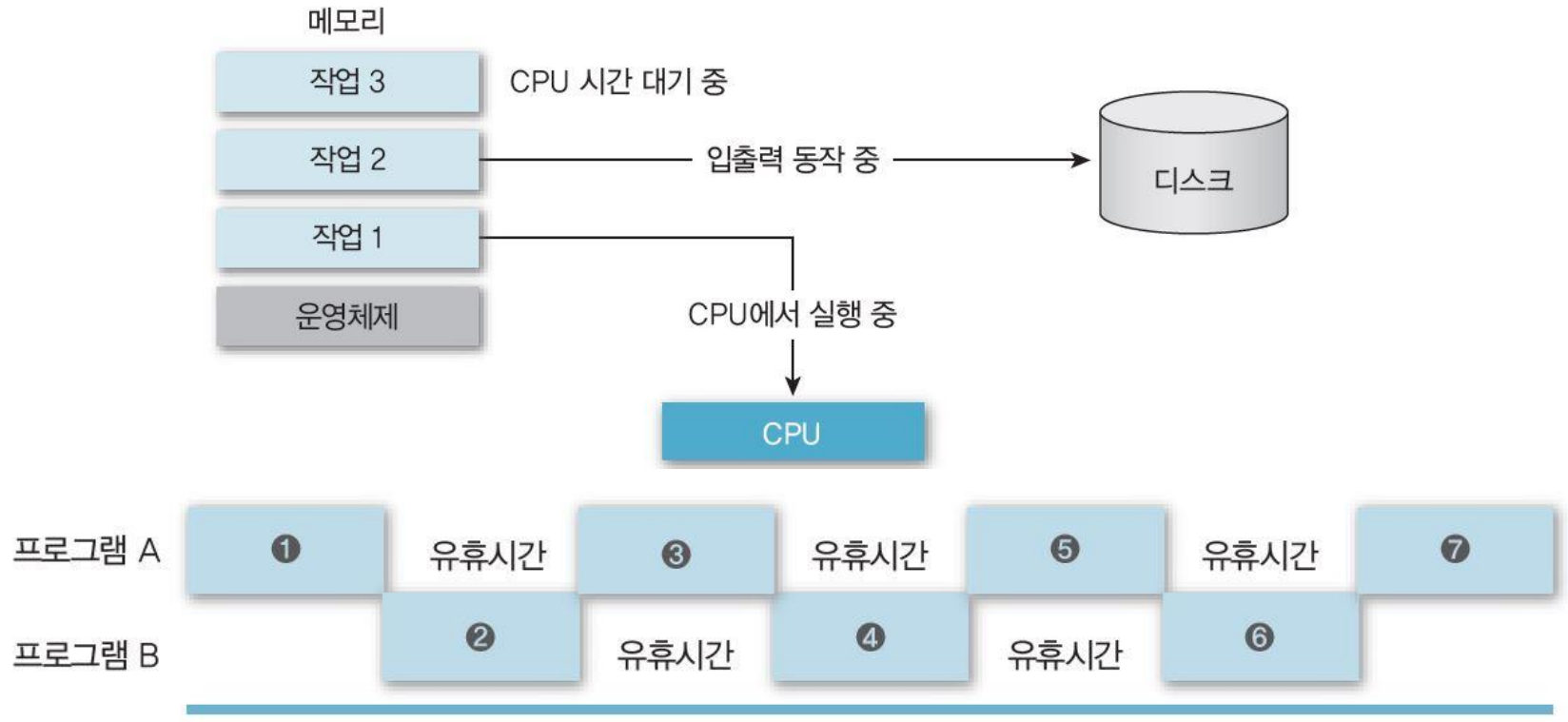
일괄 처리 시스템 Batch Processing System

- **일정량 또는 일정 기간 동안 데이터를 모아서 한꺼번에 처리하는 방식**
  - 초기 컴퓨터 시스테에서 사용된 형태
  - 급여 계산, 지불 계산, 연말 결산 등의 업무에 사용

- **문제점**
  - 프로세스가 다른 작업을 하고 있을 때는 입출력 작업이 불가능하여 프로세서와 메인 메모리의 활용도가 떨어짐

다중 프로그래밍 Multi-Programming 시스템

- **입출력 동작 등으로 프로세서가 유휴 상태일 때 실행 중인 다른 프로그램으로 전환하여 프로세서를 사용할 수 있도록 동작**
  - 하나의 CPU와 주기억장치를 이용하여 여러 개의 프로그램을 동시에 처리하는 방식
  - 높고 효율적인 프로세서 사용률(효율적인 운영) 증가
  - 많은 사용자의 프로그램이 거의 동시에 프로세서를 할당받는 듯한 느낌



시분할 시스템TSS, Time Sharing System (1)

- **여러 사용자에게 짧은 간격으로 프로세서 번갈아 할당, 마치 자기 혼자 프로세서를 독점하고 있는 양 착각하게 하여 여러 사용자가 단일 컴퓨터 시스템을 동시 사용 가능**
  - 다중 프로그래밍을 논리적으로 확장한 개념, 프로세서가 다중 작업을 교대로 수행
  - 다수의 사용자가 동시에 컴퓨터의 자원을 공유할 수 있는 기술
  - Round-Robin 방식이라고도 한다

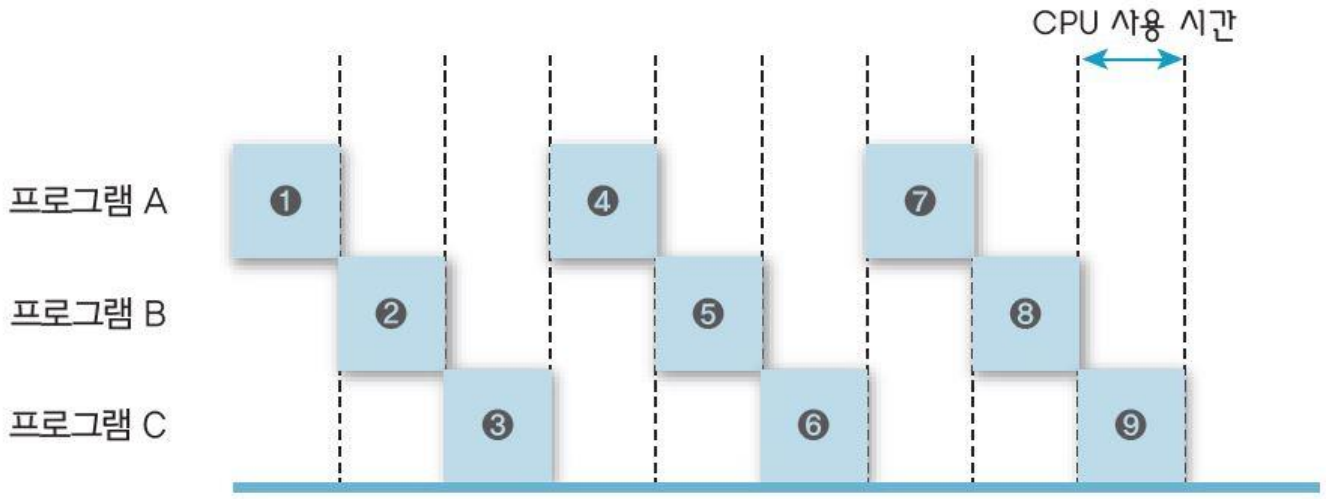


그림 2-14 시분할 시스템의 처리 방법 예

시분할 시스템TSS, Time Sharing System (2)

- **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	• 빠른 응답 제공 • 소프트웨어의 중복 회피 가능 • 프로세서 유휴시간 감소
단점	• 신뢰성 문제 • 보안 의문 및 사용자 프로그램과 데이터의 무결성 • 데이터 통신의 문제

다중 처리multiprocessing 시스템

- **여러 개의 CPU와 하나의 주기억장치를 이용하여 여러 개의 프로그램을 동시에 처리하는 방식**
  - 여러 프로세서와 시스템 버스, 메모리와 주변장치 등 공유
  - 빠르고, 프로세서 하나가 고장 나도 다른 프로세서 사용하여 작업 계속, 신뢰성 높음

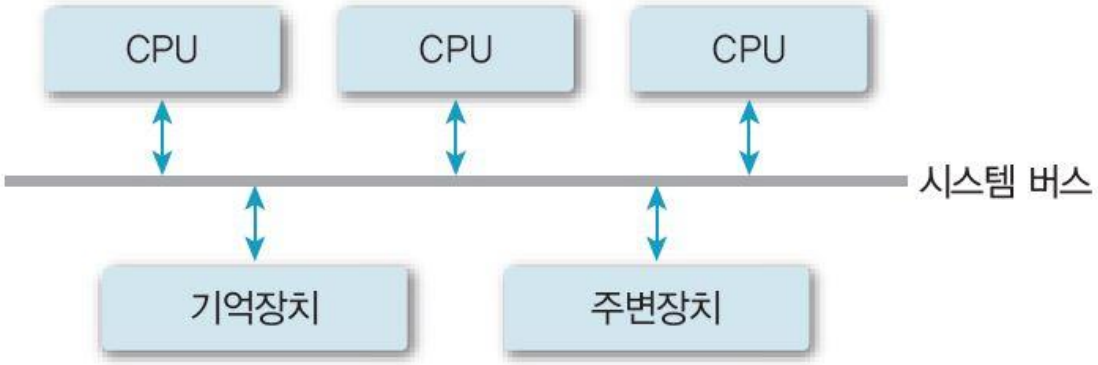


그림 2-15 다중 처리 시스템

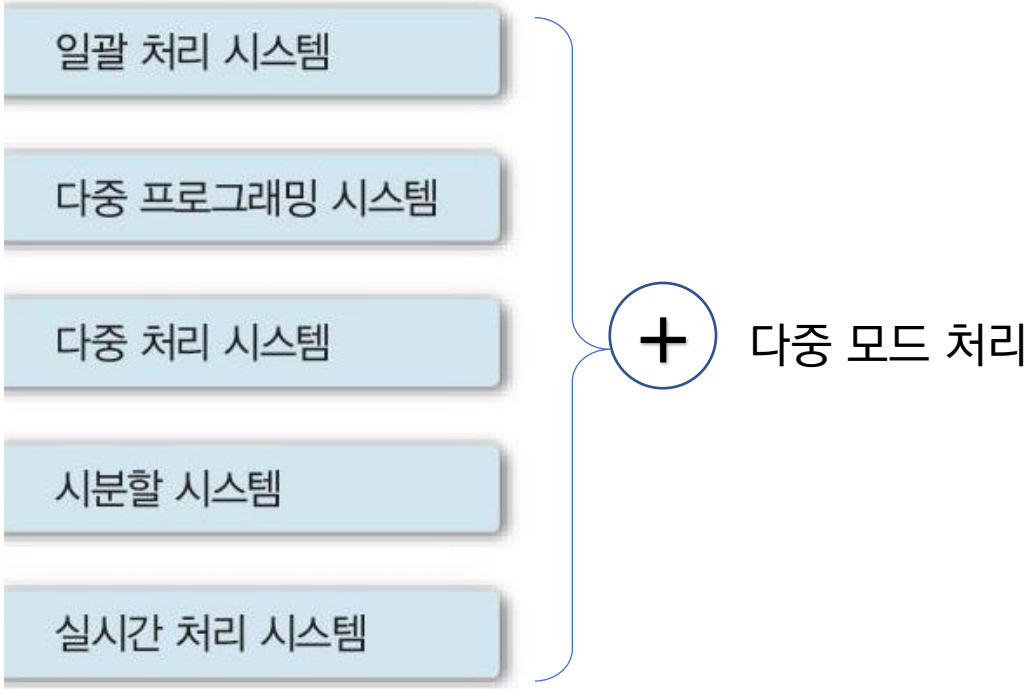
실시간 처리 시스템real time processing system

- **데이터 발생 즉시, 또는 데이터 처리 요구가 있는 즉시 처리하여 결과를 출력하는 방식**
  - 반응 시간을 고정 → 고정된 반응시간 내에 처리할 수 있도록 Processor 운용

- **실시간 처리 시스템의 두 가지 유형**
  - 경성 실시간 처리 시스템hard 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 시스템soft real time processing system
    - 작업 실행에서 시간 제약 조건은 있으나, 이를 지키지 못해도 전체 시스템에 치명적인 영향을 미치지 않는 시스템
    - 동영상 재생 : 초당 일정 프레임 이상의 영상을 재생해야 한다는 제약이 있으나, 일부 프레임을 건너뛰어도 동영상을 재생 시스템에는 큰 영향을 미치지 않음

다중 모드 처리 Multi-Mode Processing

- 일괄 처리 시스템, 시분할 시스템, 다중 처리 시스템, 실시간 처리 시스템을 한 시스템에서 모두 제공하는 방식



분산 처리 시스템distributed processing system

- **하나의 프로그램(작업)을 네트워크로 연결된 여러 개의 컴퓨터(프로세서)에 분산하여 동시에 실행**
  - 시스템마다 독립적인 운영체제와 메모리로 운영, 필요 시 통신하는 시스템
  - 데이터를 여러 위치에서 처리·저장, 여러 사용자가 공유
  - 사용자에게는 중앙집중식 시스템처럼 보이는데, 다수의 독립된 프로세서에서 실행

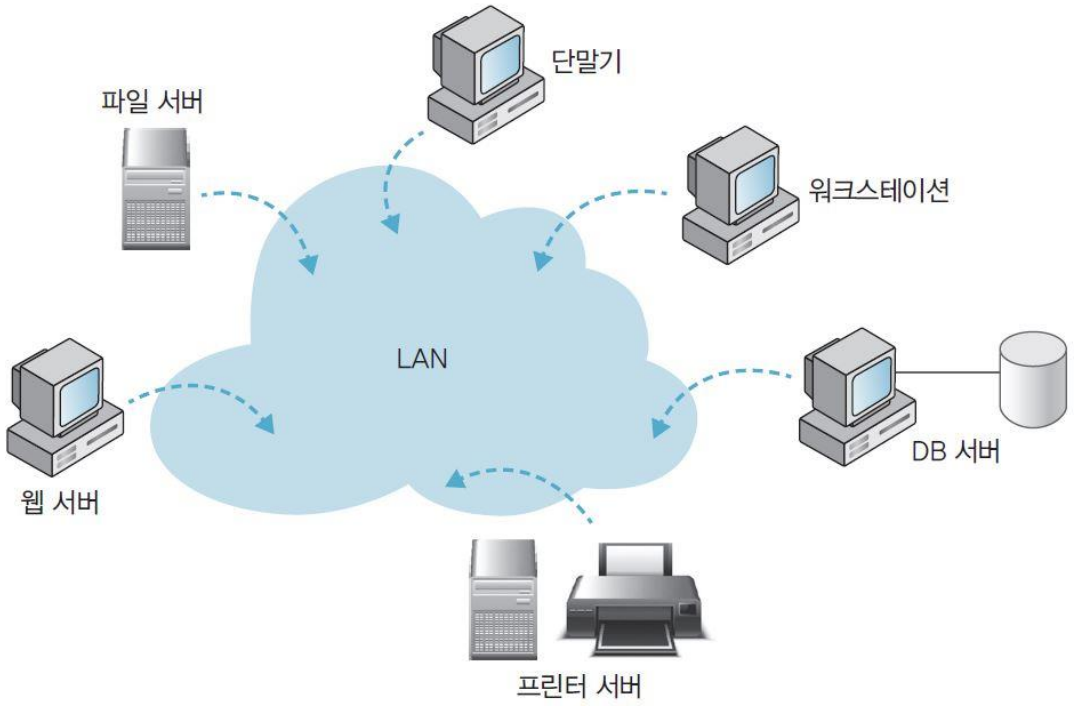
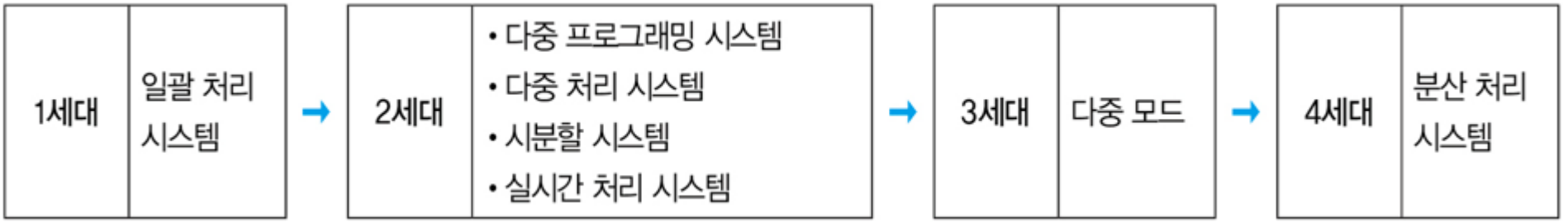


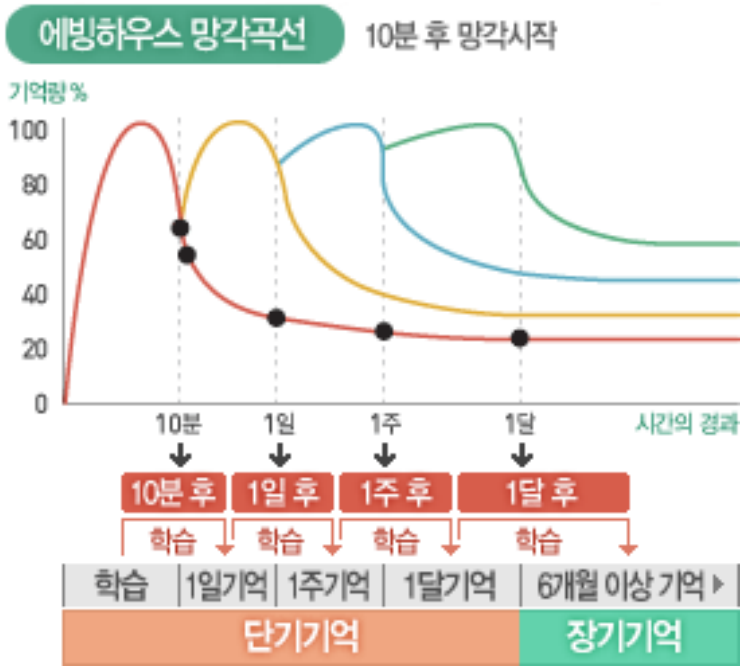
그림 2-16 분산 처리 시스템

운영체제 운용 기법의 발달 과정

기출 (22.4)



망각 곡선 되살리기


에빙하우스 망각곡선
10분 후 망각시작
기억량 %
100
80
60
40
20
0
10분
1일
1주
1달
시간의 경과
10분 후
1일 후
1주 후
1달 후
학습
학습
학습
학습
학습
1일기억
1주기억
1달기억
6개월 이상 기억
단기기억
장기기억