

5주차 1차시

## 운영체제(1)

컴퓨터 시스템과 사용자 간의 중개자\_ 운영 체제의 개념과 동작 원리

1 운영체제의 개요

2 범용 운영체제

3 프로세스 관리



## 운영체제(1)

### ◆ 학습목표

- 운영체제가 무엇이고 처리 방식에 따라 어떻게 나뉘는지 알아본다.
- 대표적인 범용 운영체제의 종류를 살펴본다.

# 1. 운영체제의 개요

## 1) 운영체제의 정의와 목적

### ◆ 운영체제의 정의

하드웨어와 사용자 사이에서 인터페이스를 제공하고, 한정된 시스템 자원을 더욱 효율적으로 관리하여 컴퓨터의 성능을 향상시키는 시스템 소프트웨어

### ◆ 운영체제의 목적

- 처리량 향상
- 응답 시간 단축
- 신뢰성 향상
- 유용성 증대

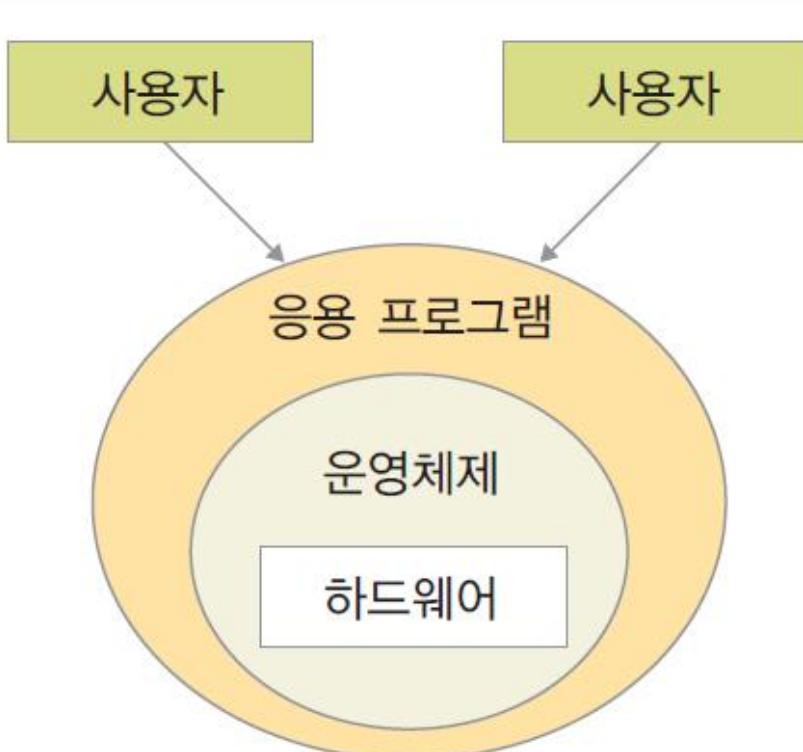
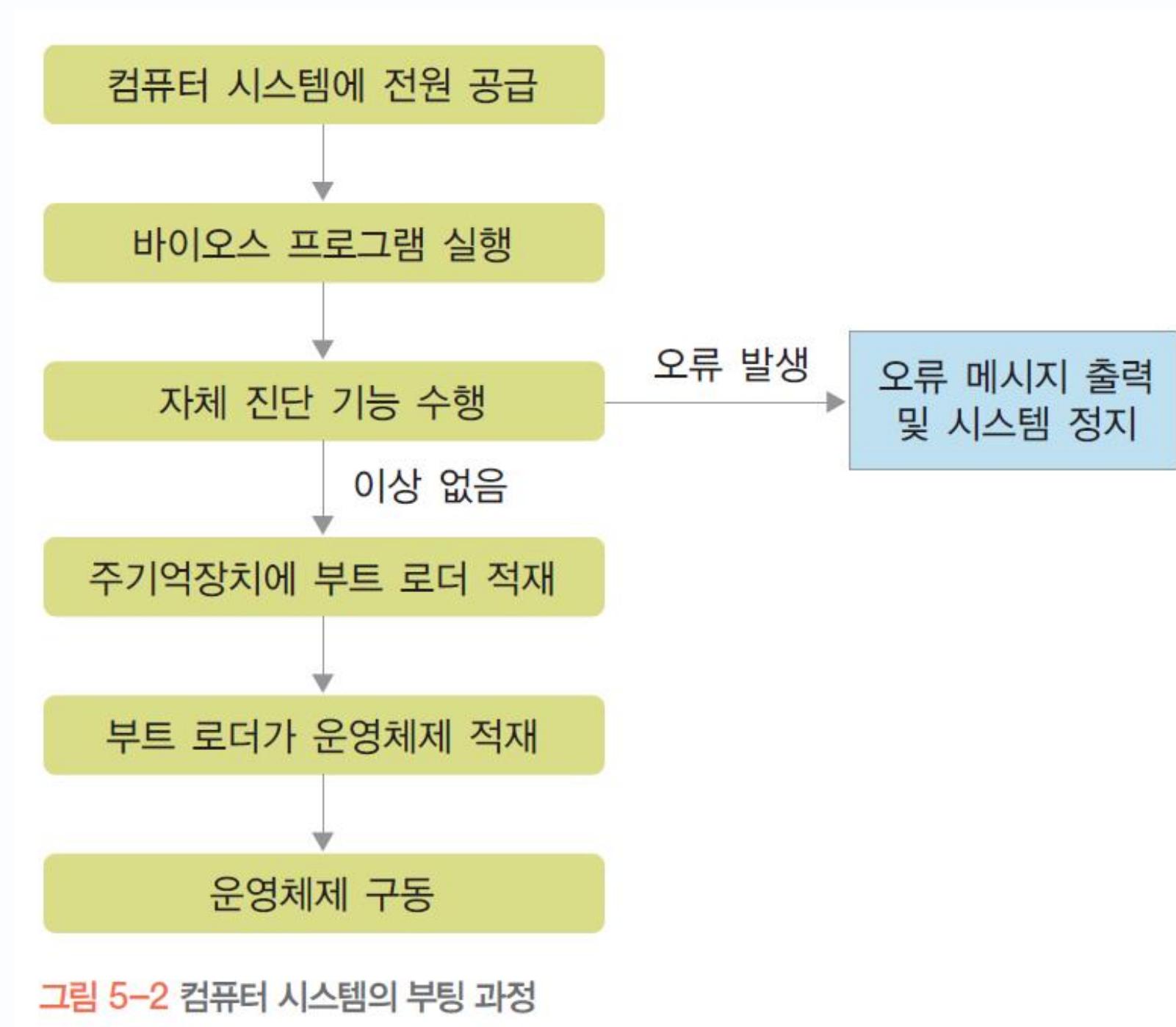


그림 5-1 컴퓨터 시스템의 구성

# 1. 운영체제의 개요

## 2) 운영체제의 구동

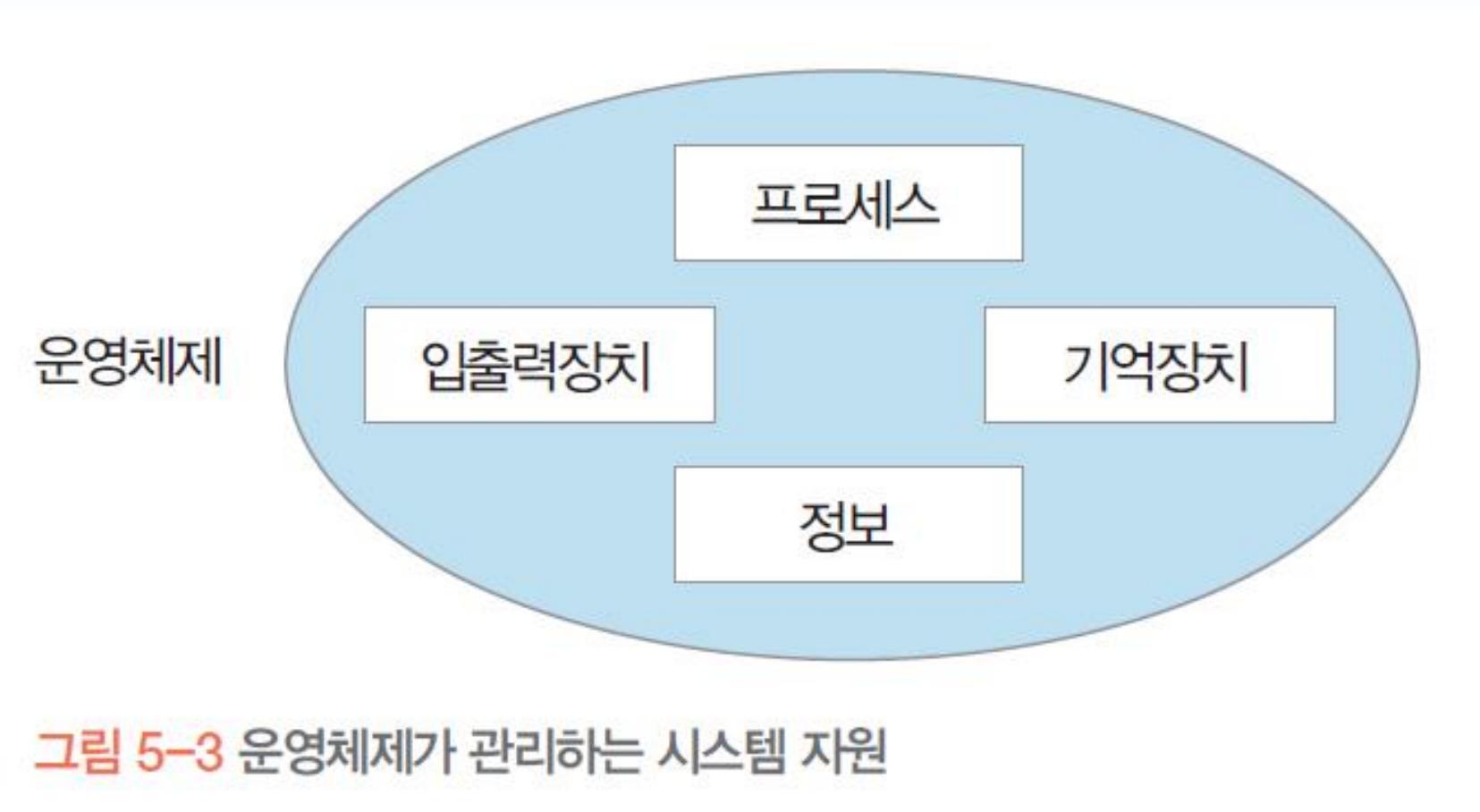
### ◆ 컴퓨터 시스템의 부팅 과정



# 1. 운영체제의 개요

## 3) 운영체제의 역할

- ◆ 프로세스 관리
- ◆ 기억장치 관리
- ◆ 입출력장치 관리
- ◆ 정보 관리

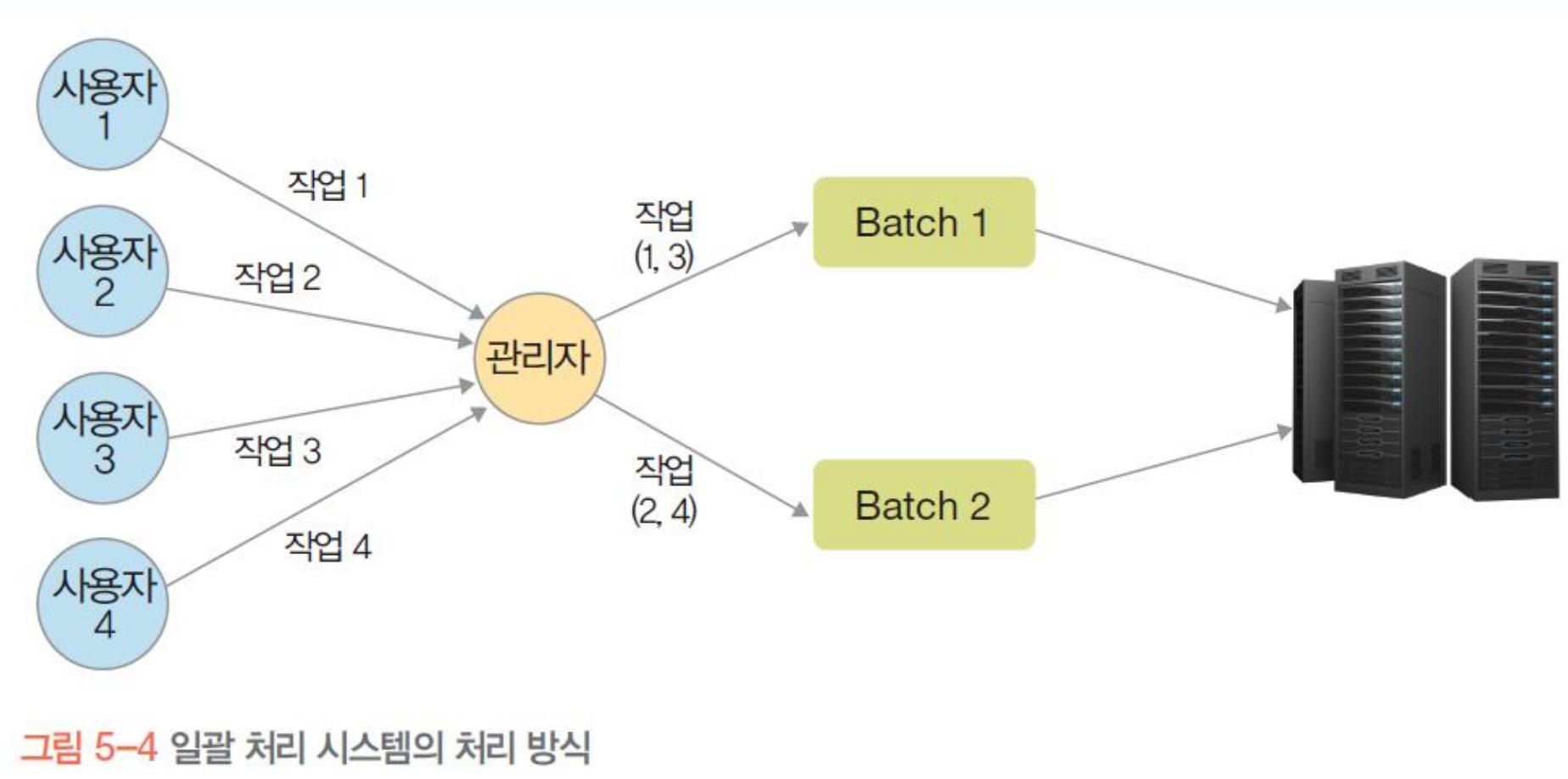


# 1. 운영체제의 개요

## 4) 운영체제의 분류

### ◆ 일괄 처리 시스템

- 자료가 발생할 때마다 즉시 처리하지 않고 입력되는 자료를 일정 기간 또는 일정량이 될 때까지 모아두었다가 한꺼번에 처리
- (예) 전기, 수도, 가스 요금의 계산, 수학능력시험 채점, 급여 계산 등



# 1. 운영체제의 개요

## 4) 운영체제의 분류

### ◆ 다중 프로그래밍 시스템

- 다수의 프로그램을 주기억장치에 적재시킨 후 CPU를 번갈아 가며 사용
- 하나의 프로그램이 CPU를 사용하다가 입출력연산을 수행하기 위해 입출력장치를 사용하면, 그 동안 다른 프로그램이 CPU를 사용하는 방법

# 1. 운영체제의 개요

## 4) 운영체제의 분류

### ◆ 다중 처리 시스템

- 컴퓨터에 두 개 이상의 CPU를 탑재하여 동시에 처리하는 방식
- 목적 : 컴퓨터 시스템의 처리 능력 증대와 신뢰성 향상

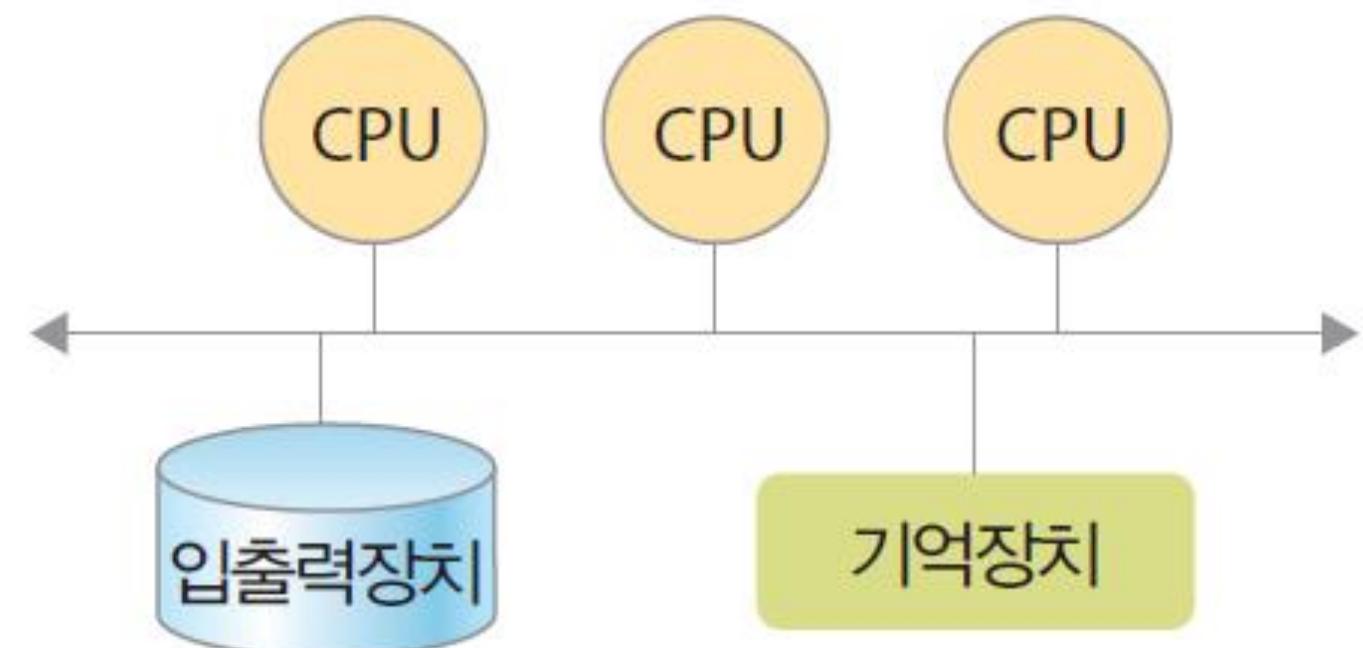


그림 5-6 다중 처리 시스템의 구조

# 1. 운영체제의 개요

## 4) 운영체제의 분류

### ◆ 시분할 시스템

- 각 사용자에게 일정한 CPU 시간을 할당
- 주어진 시간 동안 컴퓨터와 대화 형식으로 프로그램을 실행하는 방식
- 각 사용자에게 컴퓨터 시스템을 독점적으로 사용하고 있다는 느낌을 줌

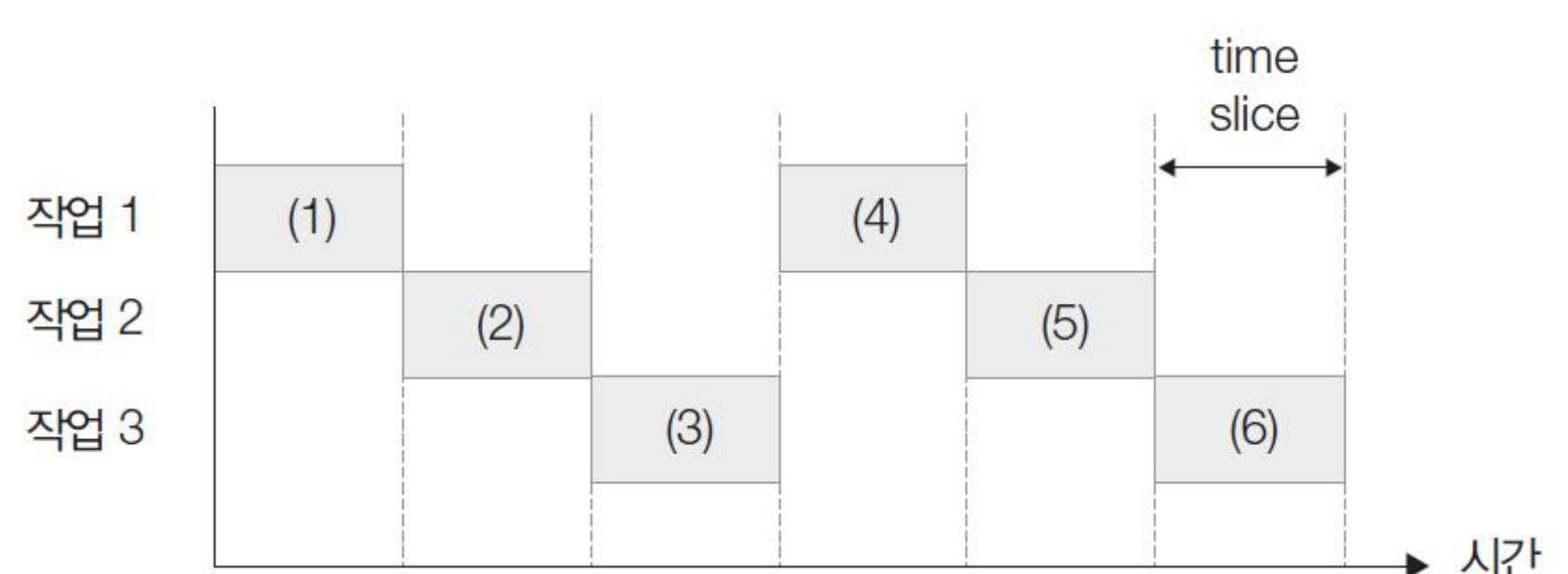


그림 5-7 시분할 시스템의 처리 방식

# 1. 운영체제의 개요

## 4) 운영체제의 분류

### ◆ 실시간 시스템

- 입력된 데이터를 즉시 처리한 후 바로 결과를 보냄
- 작업의 요청에서부터 수행 결과를 얻기까지 시간제약이 존재
- 제약의 엄격함에 따라 경성 실시간 시스템과 연성 실시간 시스템으로 구분

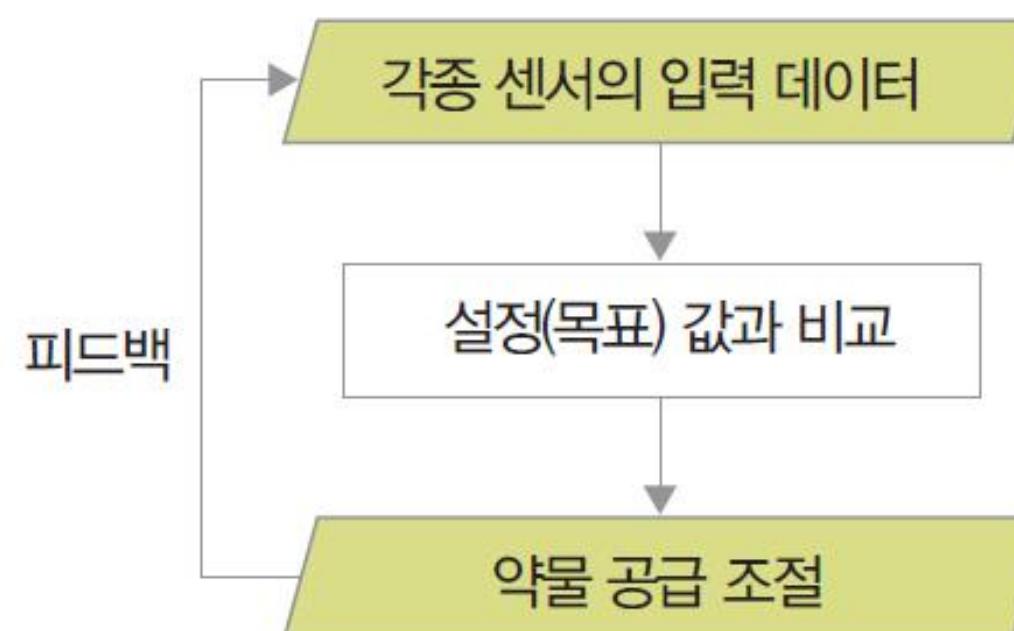


그림 5-8 실시간 시스템(환자 모니터링 시스템)의 처리 방식

# 1. 운영체제의 개요

## 4) 운영체제의 분류

### ◆ 분산 처리 시스템

- 네트워크를 통해 연결된 다수의 컴퓨터 시스템에 작업과 자원을 분산시켜 처리
- 설계 목적은 자원 공유, 연산 속도와 신뢰성 향상, 통신 기능 활용

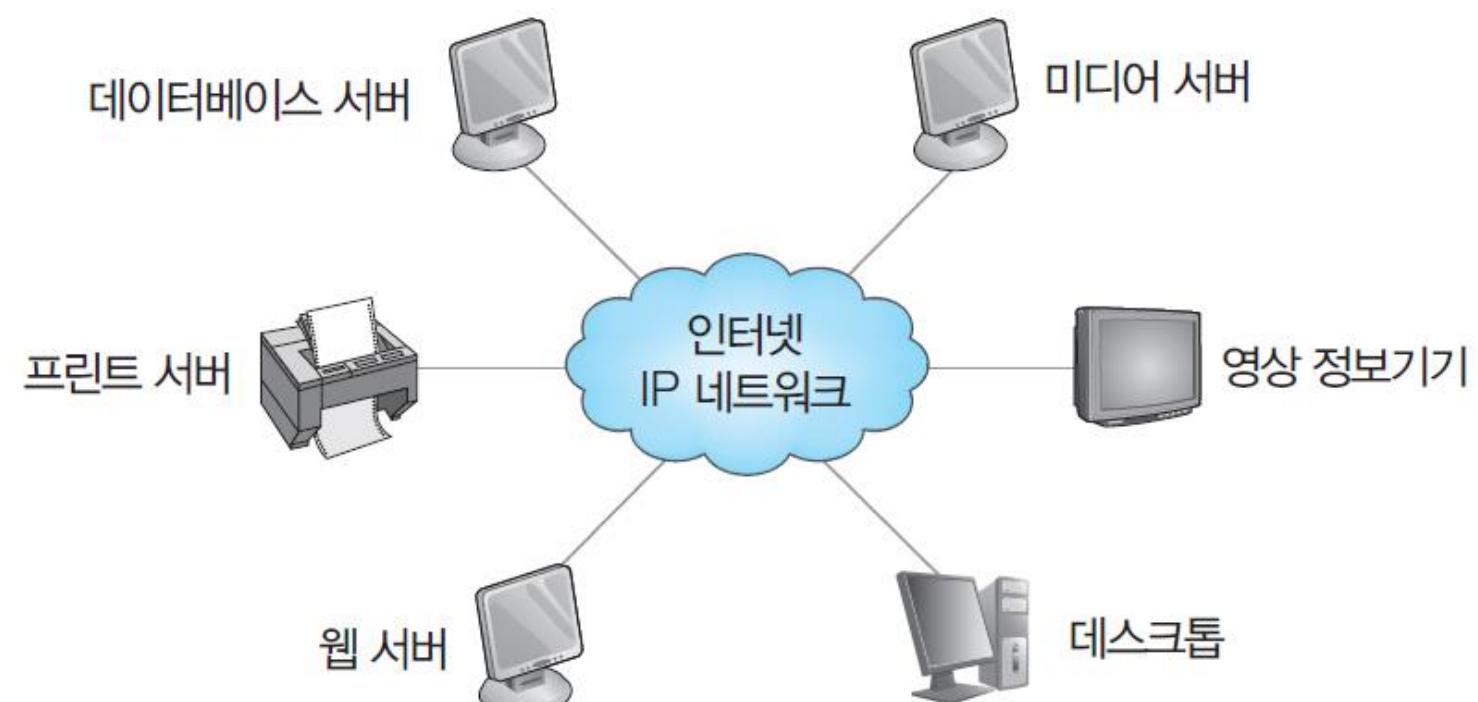


그림 5-9 분산 처리 시스템의 처리 방식

## 2. 범용 운영체제

### 1) 유닉스

- ◆ 1969년 미국 벨 연구소의 켄 톰슨에 의해 PDP-7 시스템의 운영체제로 처음 개발
- ◆ 프로그램의 소규모 모임으로 시작
- ◆ 현재는 소규모 내장 컨트롤러에서 메인 프레임 컴퓨터까지 다양한 영역에서 수많은 응용 프로그램을 구동시키는 개방형 표준 운영체제로 발전

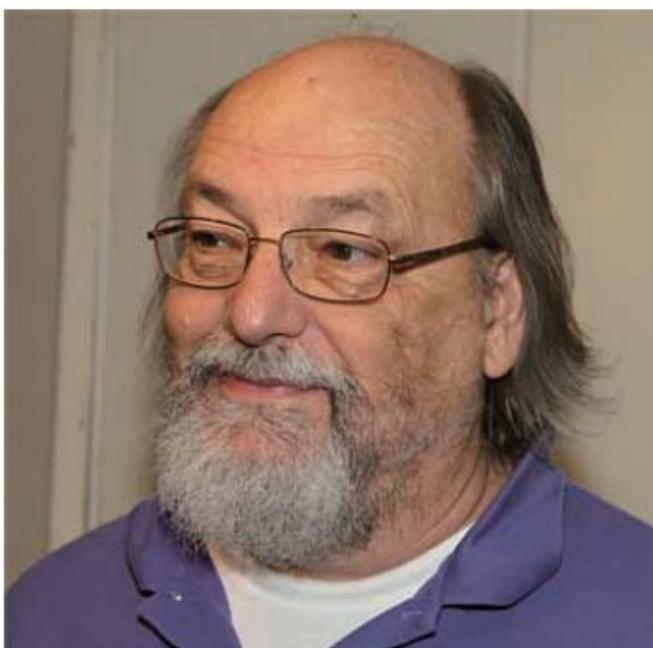


그림 5-10 켄 톰슨(왼쪽)과 PDP-7 시스템(오른쪽)

## 2. 범용 운영체제

### 1) 유닉스

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 유ти리티 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

## 2. 범용 운영체제

### 2) 리눅스

- ◆ 1991년 핀란드 헬싱키 대학의 학생 리누스 토발즈에 의해 유닉스 호환 운영체제로 개발
- ◆ 오픈 소스 정책을 따름

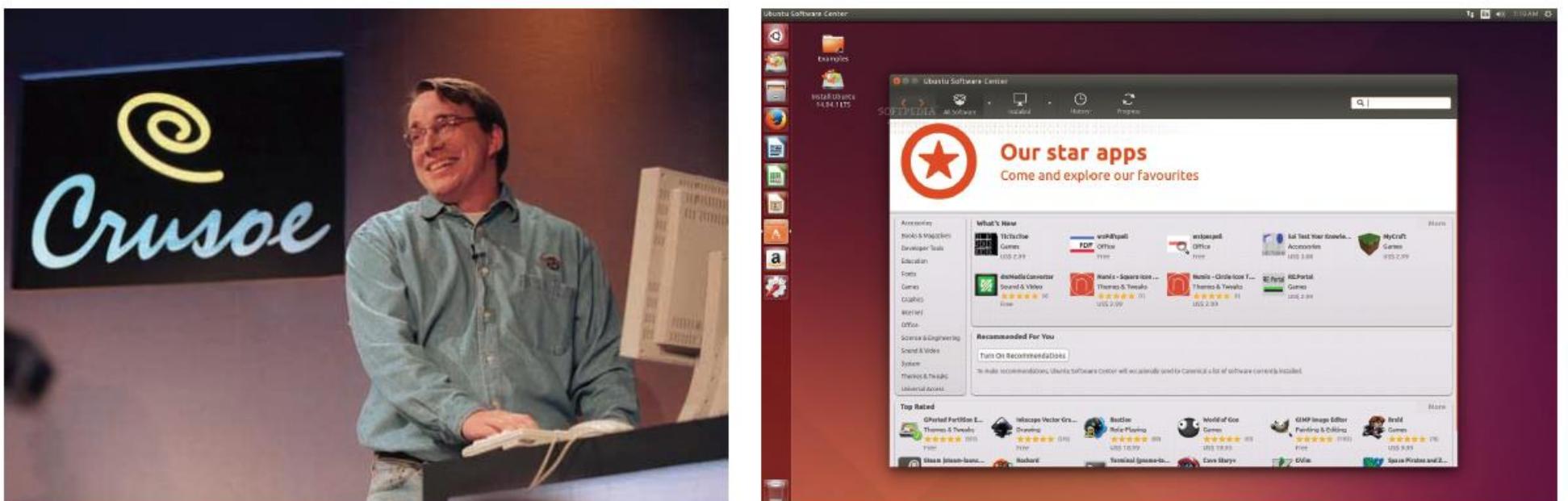


그림 5-11 리누스 토발즈(왼쪽)와 우분투 리눅스의 실행 화면(오른쪽)

## 2. 범용 운영체제

### 3) 리눅스의 역사

- ◆ 1991년 핀란드 헬싱키 대학 학생이던 리누스 토발스(Linus Torvalds)가 대형 기종에서나 작동하던 OS인 UNIX를 386 기종의 PC에서도 작동할 수 있게 만듬
- ◆ 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

## 2. 범용 운영체제

### 4) 리눅스의 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 배포판들

## 2. 범용 운영체제

### 5) 리눅스의 단점

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

## 2. 범용 운영체제

### 6) 윈도우

- ◆ 애플의 그래픽 사용자 인터페이스 운영체제인 맥 OS에 대항하기 위해 개발한 운영체제
- ◆ 자사의 MS-DOS에 다중 작업 기능과 GUI 환경을 추가

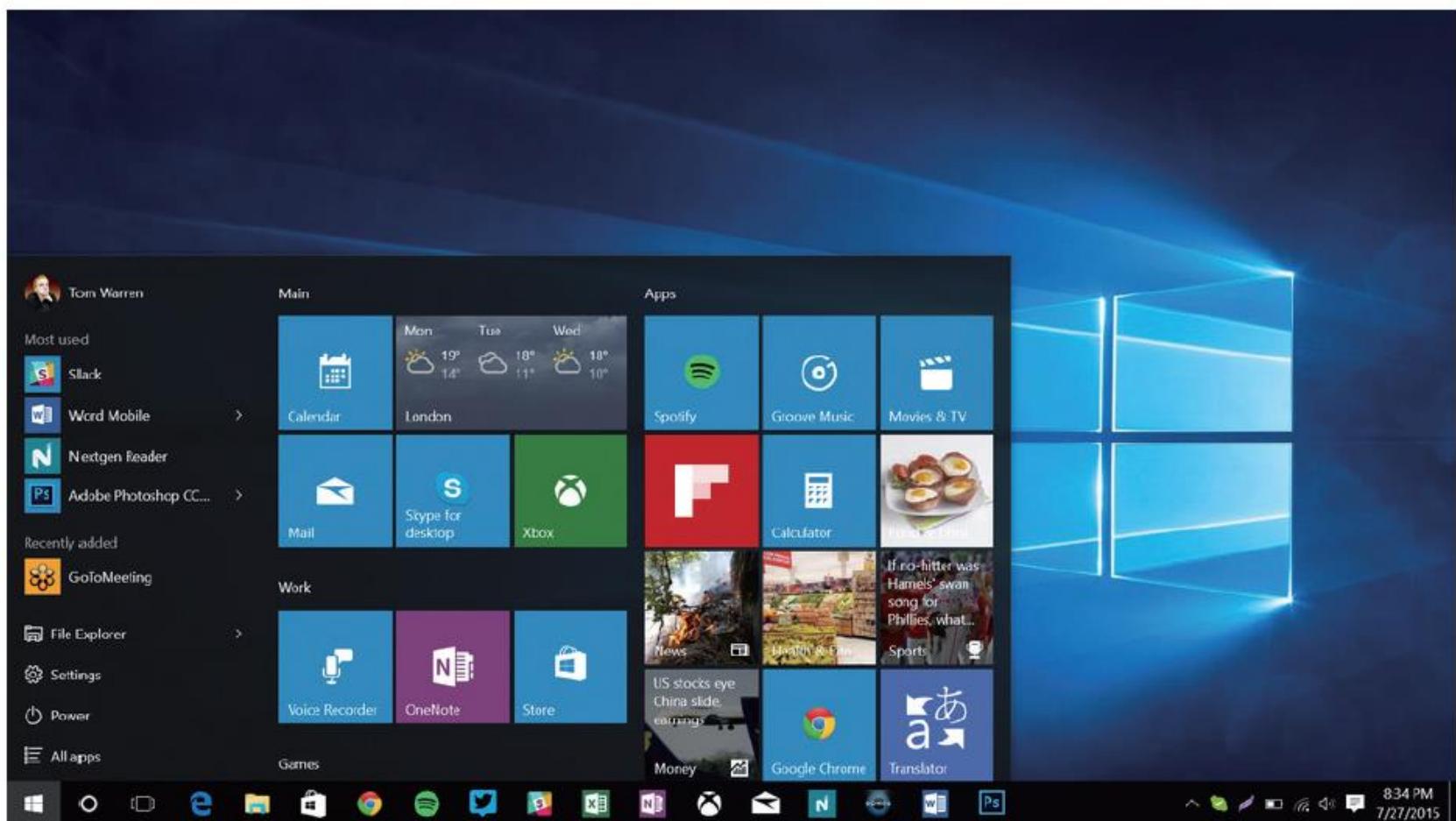


그림 5-12 윈도우 10의 실행 화면

## 2. 범용 운영체제

### 6) 윈도우

표 5-1 윈도우의 버전별 특징

버전명	출시연도	특징
윈도우 1.0	1985	<ul style="list-style-type: none"> <li>MS-DOS의 확장판</li> </ul>
윈도우 3.0/3.1	1990/1992	<ul style="list-style-type: none"> <li>사용자 인터페이스 개선</li> <li>강력한 다중 작업 기능 제공</li> </ul>
윈도우 NT	1993	<ul style="list-style-type: none"> <li>NT 커널 기반의 전문가용 운영체제</li> <li>NTFS 파일 시스템, 강력한 보안 기능, 네트워킹 모듈 제공</li> </ul>
윈도우 95	1995	<ul style="list-style-type: none"> <li>진정한 의미의 32비트 운영체제</li> </ul>
윈도우 98	1998	<ul style="list-style-type: none"> <li>FAT32 파일 시스템 지원</li> <li>개선된 PnP(Plug and Play) 기능 제공</li> <li>USB, IEEE1394, AGP(Accelerated Graphics Port) 등 지원</li> <li>다중 모니터 및 표준 전원 관리 기능(ACPI) 제공</li> </ul>
윈도우 me	2000	<ul style="list-style-type: none"> <li>UPnP(Universal PnP)와 시스템 복원 기능 추가</li> <li>9x 기반의 개인용 윈도우의 마지막 버전</li> </ul>
윈도우 2000	2000	<ul style="list-style-type: none"> <li>기업/전문가용 운영체제</li> </ul>

## 2. 범용 운영체제

### 6) 윈도우

표 5-1 윈도우의 버전별 특징

버전 명	출시 연도	특징
윈도우 XP	2001	<ul style="list-style-type: none"> <li>소비자 지향형 운영체제(NT 커널 기반+윈도우 me 편의성)</li> <li>홈 에디션과 프로페셔널 버전으로 구분</li> </ul>
윈도우 XP 미디어 센터	2003	<ul style="list-style-type: none"> <li>DVD/TV 기능(프로그램 녹화와 원격 조정 기능 지원)</li> </ul>
윈도우 서버 2003	2003	<ul style="list-style-type: none"> <li>윈도우 2000 서버 제품군에 새로운 기능과 강력한 보안 기능 추가</li> </ul>
윈도우 비스타	2006/2007	<ul style="list-style-type: none"> <li>기업용 버전과 개인용 버전을 별도로 출시</li> <li>윈도우 XP보다 강화된 보안 기능 제공</li> </ul>
윈도우 서버 2008	2008	<ul style="list-style-type: none"> <li>윈도우 서버 2003의 후속작(윈도우 비스타 기반)</li> </ul>
윈도우 7	2009	<ul style="list-style-type: none"> <li>다양한 에디션 및 32/64비트 버전 출시</li> <li>윈도우 XP와 비스타 기반의 프로그램 실행 가능</li> </ul>
윈도우 서버 2008 R2	2009	<ul style="list-style-type: none"> <li>윈도우 서버 2008의 후속작(윈도우 7 기반)</li> </ul>
윈도우 8/8.1	2012/2013	<ul style="list-style-type: none"> <li>인텔/AMD 외에도 ARM 프로세서 지원(윈도우 RT)</li> <li>터치스크린 기능 탑재</li> </ul>
윈도우 서버 2012	2012	<ul style="list-style-type: none"> <li>윈도우 서버 2008 R2의 후속작(윈도우 8 기반)</li> <li>네 가지 에디션(파운데이션/에센셜/스탠다드/데이터 센터)</li> </ul>
윈도우 서버 2012 R2	2013	<ul style="list-style-type: none"> <li>윈도우 서버 2012의 후속작(윈도우 8.1 기반)</li> </ul>

## 2. 범용 운영체제

### 7) 맥 OS

- ◆ 애플의 개인용 컴퓨터인 매킨토시용으로 개발된 GUI 운영체제
- ◆ 애플의 자체 코드에 기반을 둔 클래식 시리즈와 BSD 코드를 사용하는 유닉스 계열 운영체제인 NEXTSTEP 기반의 맥 OS X로 나눔

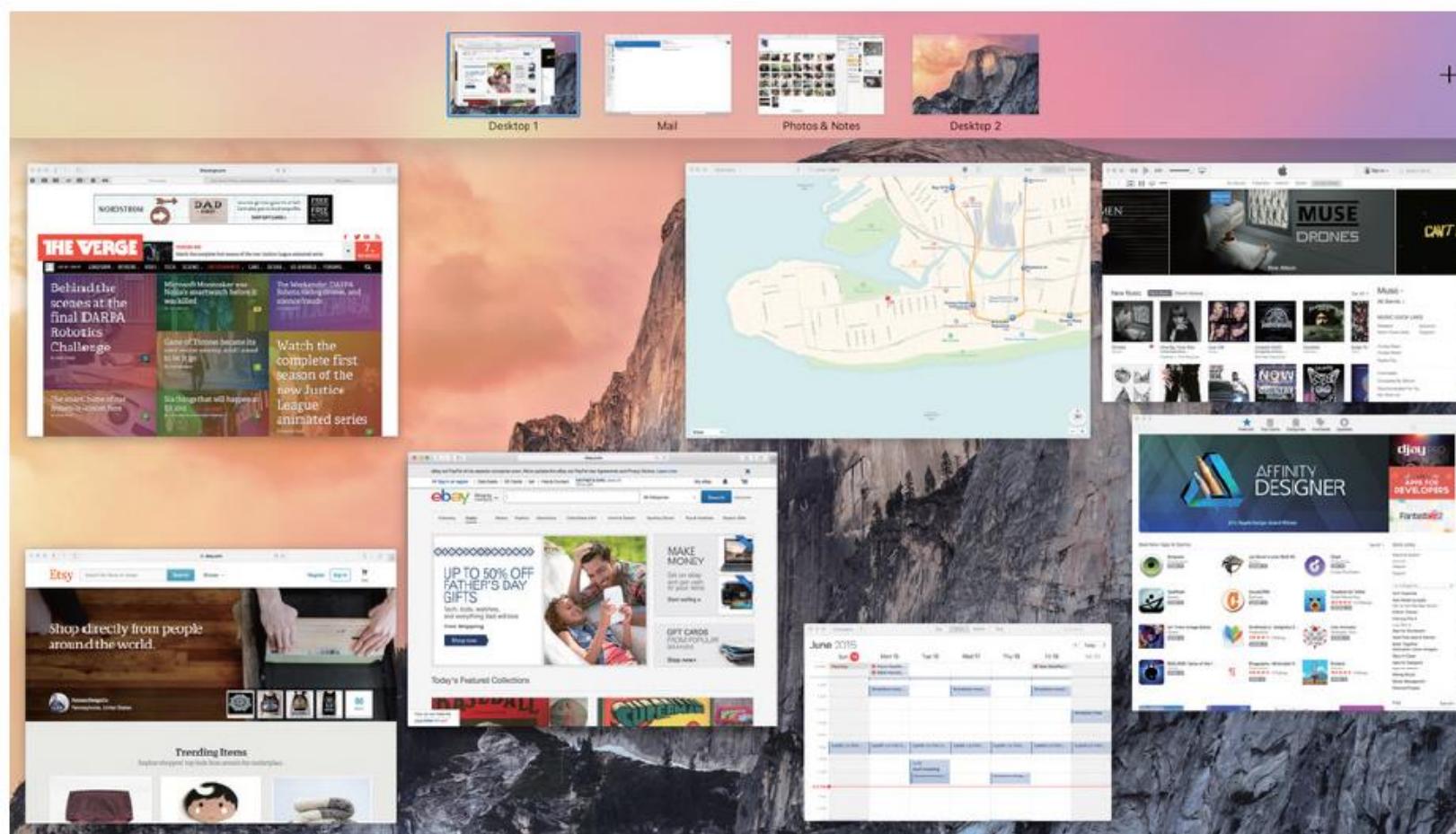


그림 5-13 맥 OS X 10.11.1의 실행 화면

### 3. 프로세스 관리

#### 1) 프로세스의 개념

- ◆ 실행중인 프로그램
- ◆ 프로그램은 디스크에 저장된 파일과 같은 수동적인 개체
- ◆ 프로세스는 프로그램 카운터(PC)를 가진 능동적인 개체
- ◆ 프로그램이 주기억장치에 적재되어 처리를 시작할 때 비로소 프로세스라는 활동적인 개체가 됨

### 3. 프로세스 관리

#### 2) 프로세스 상태

- ◆ 초기 상태(new state)
- ◆ 준비 상태(ready state)
- ◆ 실행 상태(running state)
- ◆ 대기 상태(blocked state)
- ◆ 종료 상태(exit state)

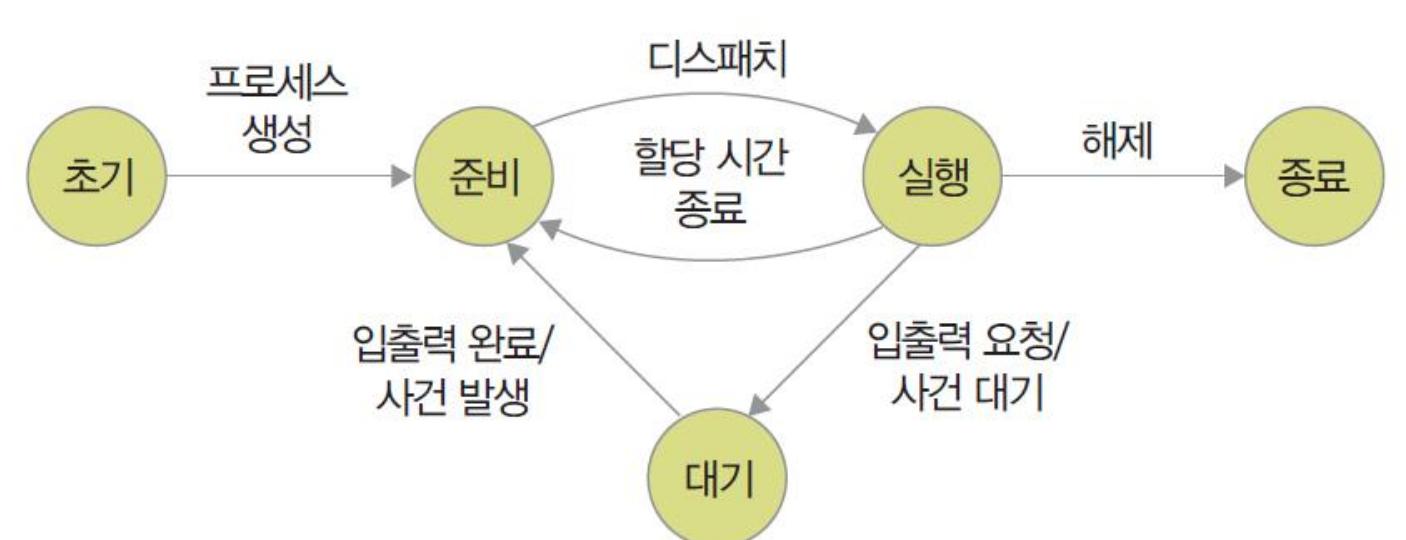


그림 5-14 프로세스 상태 전이도

### 3. 프로세스 관리

#### 3) 프로세스 제어 블록

- ◆ 프로세스에 대한 정보를 운영체제에 제공하는 자료 구조
- ◆ 프로세스가 생성될 때 만들어졌다가 프로세스가 실행을 마치면 삭제됨

포인터	프로세스 상태
	프로세스 식별자
	프로그램 카운터
	레지스터
	기억장치 관리 정보
	개방된 파일 리스트
	기타 정보

그림 5-15 프로세스 제어 블록(PCB)

### 3. 프로세스 관리

#### 4) 병행 프로세스

- ◆ 두 개 이상의 연관된 프로세스가 동시에 실행되는 것
- ◆ 두 개 이상의 프로세스들이 병행 처리 상태에 있으면 예측 불가능한 결과가 발생하는데, 이러한 오류를 막기 위해 다음과 같은 기법을 사용
  - 동기화(synchronization)
  - 임계 구역(critical section 또는 critical region)
  - 상호 배제(mutual exclusion)

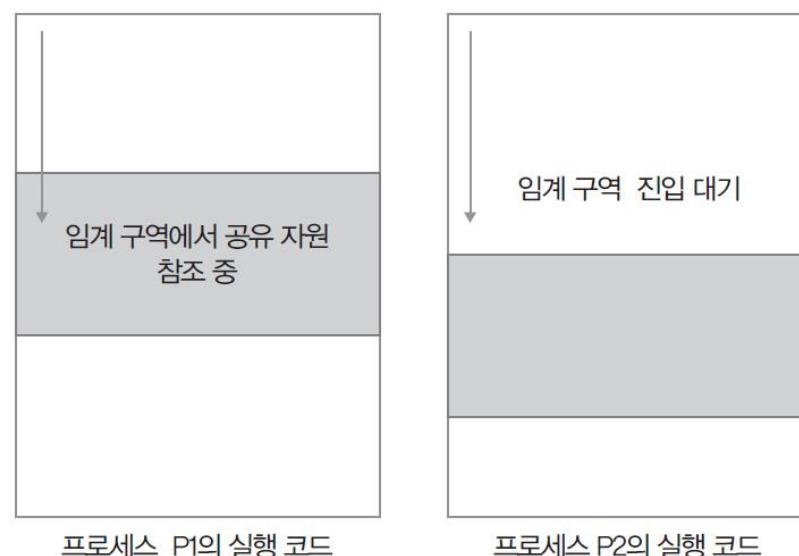


그림 5-16 임계 구역

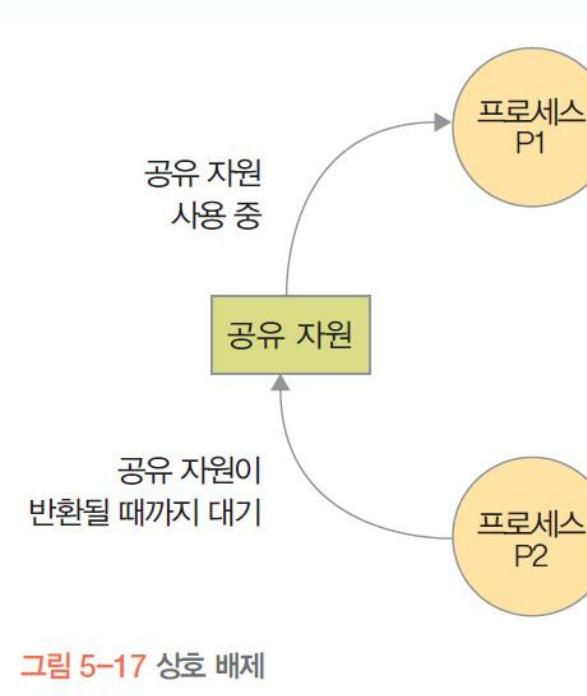


그림 5-17 상호 배제

### 3. 프로세스 관리

#### 5) 교착 상태

- ◆ 다중 프로그래밍 시스템에서 하나 이상의 프로세스가 절대 일어나지 않을 특정 사건을 기다리고 있는 상태

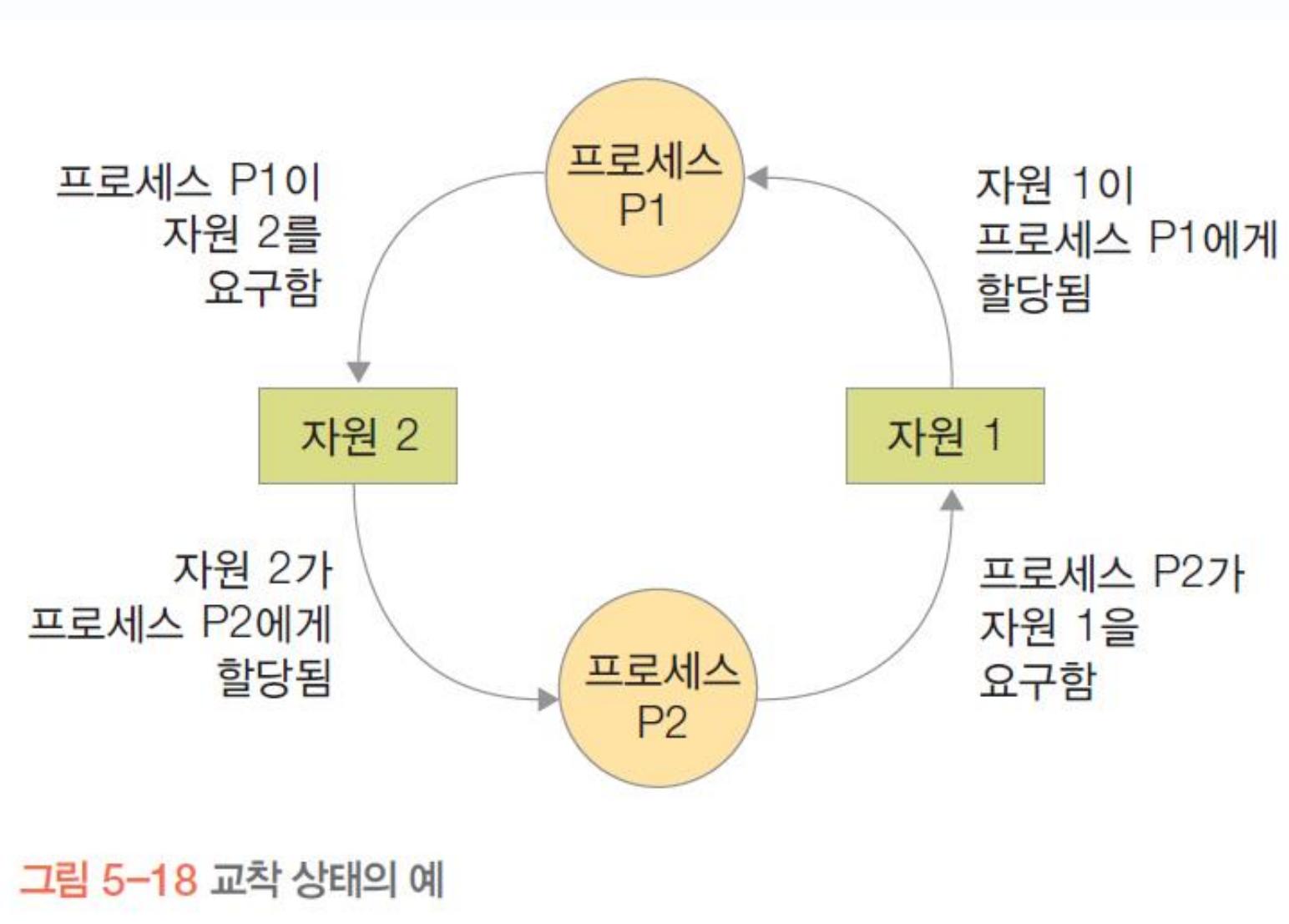


그림 5-18 교착 상태의 예

### 3. 프로세스 관리

#### 5) 교착 상태

##### ◆ 교착 상태의 발생 조건

- 상호 배제(mutual exclusion)
- 보유와 대기(hold and wait)
- 비선점(nonpreemption)
- 환형 대기(circular wait)

##### ◆ 교착 상태의 해결 방안

- 교착 상태 예방(deadlock prevention)
- 교착 상태 회피(deadlock avoidance)
- 교착 상태 탐지(deadlock detection)
- 교착 상태 회복(deadlock recovery)



## 요약

### ◆ 운영체제

- **개념:** 사용자가 컴퓨터 시스템을 편리하게 사용할 수 있도록 인터페이스를 제공하고, 한정된 시스템 자원을 효율적으로 관리하여 컴퓨터의 성능을 향상시키는 시스템 소프트웨어이다.
- **역할:** 운영체제는 프로세스, 기억장치, 입출력장치, 정보 등의 시스템 자원을 관리하는 역할을 한다.
- **분류:** 운영체제는 작업 처리 방식에 따라 일괄처리 시스템, 다중 프로그래밍 시스템, 다중 처리 시스템, 시분할 시스템, 실시간 시스템, 분산처리 시스템으로 나뉜다.



## 요약

### ◆ 프로세스

- 개념: 실행중인 프로그램을 의미하며, 다음에 실행할 명령어를 지정하는 프로그램을 가진 능동적인 개체
- 병행 프로세서: 두개 이상의 연관된 프로세스가 동시에 시행되는 것을 말함
- 교착상태: 다중 프로그래밍 시스템에서 하나 이상의 프로세스가 아무리 기다려도 발생하지 않을 특정 사건을 기다리고 있는 상태이다.