

2022 SPRING : Operating System



Operating System

JOO-MAN KIM
JOOMKIM@PUSAN.AC.KR

DEPT. OF APPLIED IT AND ENGINEERING



Chapter 1: 서론-1

- What Operating Systems Do
- Computer-System Organization
- Computer-System Architecture
- Operating-System Structure



목표

- 컴퓨터 시스템의 기본 구성을 이해한다
- 운영체제의 주요 구성 요소를 포괄적으로 이해한다
- 다양한 유형의 컴퓨팅 환경에 대한 개요를 익힌다
- 여러 오픈 소스 운영체제를 탐색해 본다



운영체제란 무엇인가?

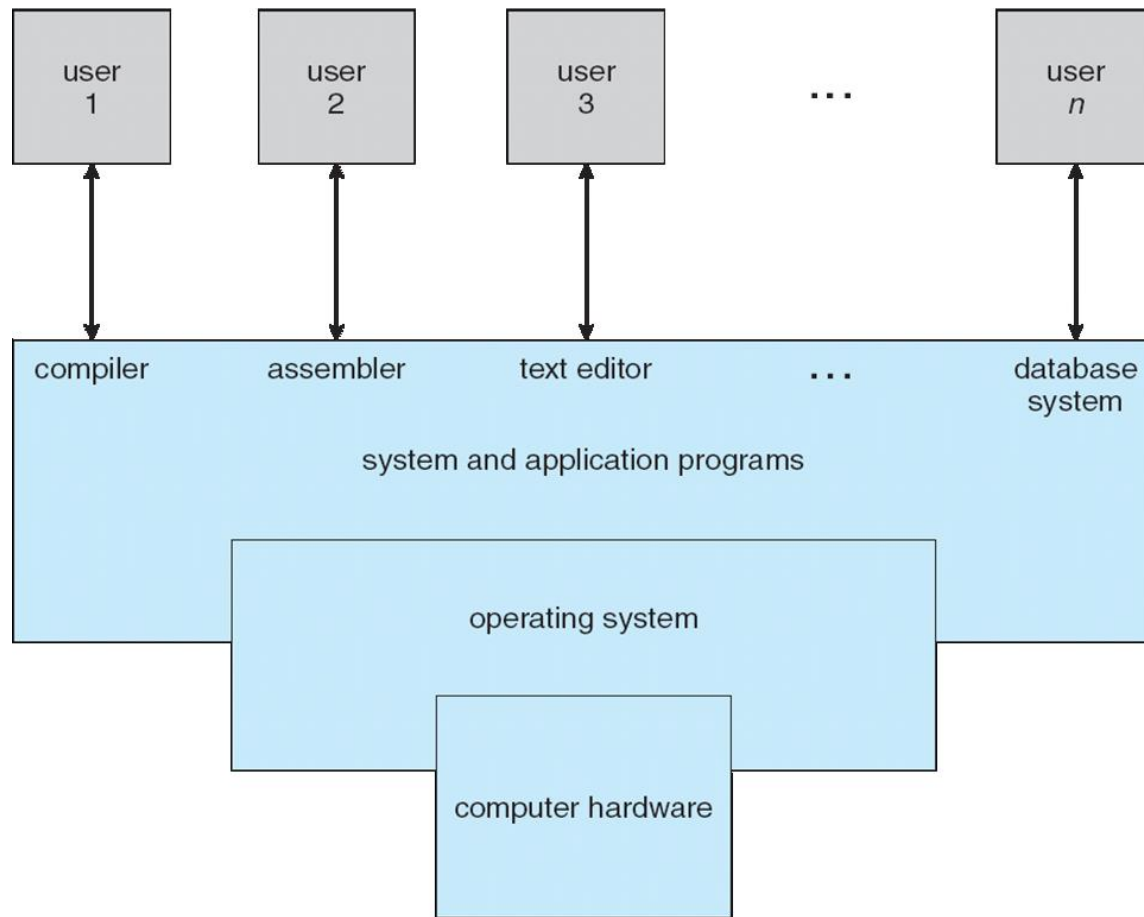
- 컴퓨터 사용자와 컴퓨터 하드웨어 사이의 중개자 역할을 하는 프로그램
- 운영체제 목표 :
 - 사용자 프로그램을 실행하고 사용자 문제를 보다 쉽게 해결할 수 있도록 하자
 - 컴퓨터 시스템을 사용하기 편리하게 만들자
 - 컴퓨터 하드웨어를 효율적으로 사용하자



컴퓨터 시스템 구조

- 컴퓨터 시스템은 4 가지 구성 요소로 나눌 수 있다
 - 하드웨어 - 기본적인 컴퓨팅 리소스를 제공한다
 - CPU, 메모리, I / O 장치
 - 운영체제
 - 다양한 응용 프로그램 및 사용자 간의 하드웨어 사용을 제어하고 조정합니다.
 - 응용프로그램 - 사용자의 컴퓨팅 문제를 해결하기 위해 시스템 자원을 사용하는 방법을 정의한다
 - 워드 프로세서, 컴파일러, 웹 브라우저, 데이터베이스 시스템, 비디오 게임
 - 사용자
 - 사람, 기계, 다른 컴퓨터

컴퓨터 시스템의 4가지 구성요소





운영체제가 하는 일

- 관점에 따라 달라진다
- 사용자는 편리함, **사용하기 쉬움** 및 **좋은 성능**을 원한다
 - **자원의 이용률**에는 관심 없다
- 그러나 **메인프레임** 또는 **미니컴퓨터**와 같은 공용 컴퓨터는 모든 사용자를 만족시켜야 한다
- **워크스테이션**과 같은 전용 시스템 사용자는 전용 자원을 가지고 있지만 종종 **서버**가 제공하는 공유 자원을 사용한다
- 휴대용 컴퓨터는 자원이 부족하고 전력 소모를 줄이고 사용의 편의성을 높이기 위해 최적화 된다
- 장치에 내장된 컴퓨터 또는 자동차에서 사용되는 일부 컴퓨터는 사용자와의 상호 작용이 극히 드물거나 없다



운영체제의 정의

- OS는 **자원 할당기**
 - 모든 자원을 관리
 - 효율과 공정한 자원 사용을 목적으로 하는 상충된 요구 사이에서 결정해야
- OS 는 **제어 프로그램**
 - 프로그램의 실행을 제어하여 컴퓨터의 잘못된 사용이나 부적합한 사용을 방지



운영체제의 정의 (계속)

- 모든 사람들이 받아들이는 정의는 없다
- “운영체제를 주문했을 때 판매자가 배송하는 모든 것 ” 이 근접한 정의 “
 - 그러나 광범위하게 달라진다
- “컴퓨터에서 항상 실행되는 유일한 프로그램”을 **커널**이라 한다
- 그 외 다른 모든 것들은
 - 시스템 프로그램(운영체제와 같이 배송됨)이거나
 - 응용프로그램



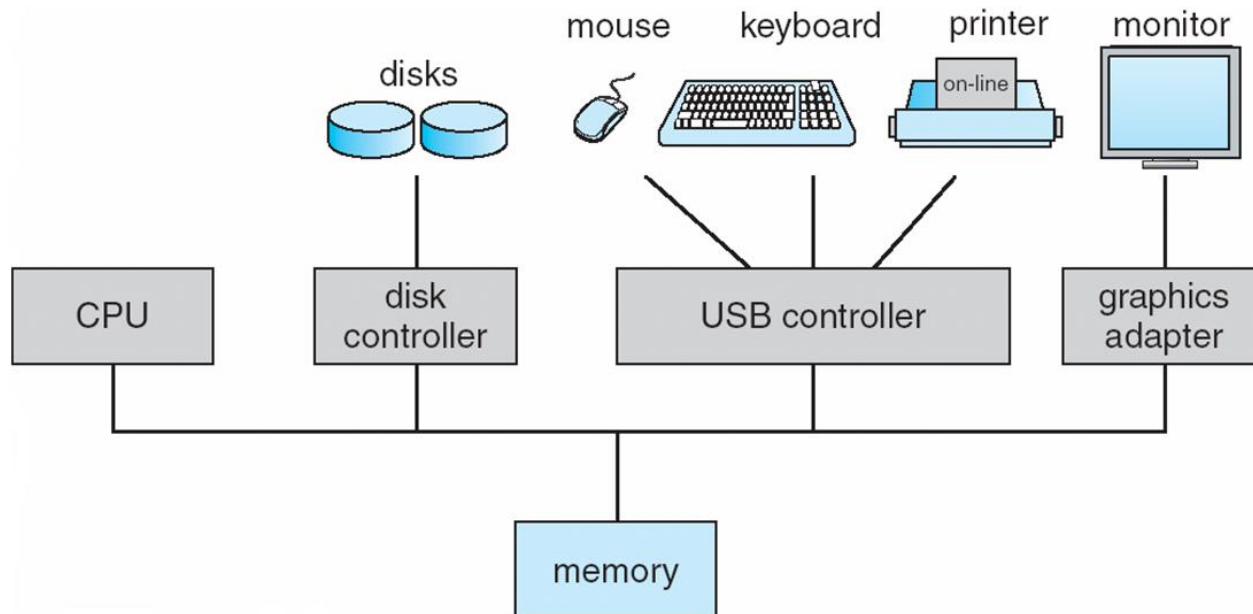
컴퓨터 시동

- **부트스트랩 프로그램**이 전원이 공급되는 시점에 혹은 다시 부트될 때 적재된다
 - 통상 ROM 또는 EPROM에 저장되며 **펌웨어**라고 불림
 - 시스템의 모든 구성요소를 초기화 한다
 - 운영체제 커널을 적재 시킨 후 커널을 실행 시킨다

컴퓨터 시스템 구성

■ 컴퓨터 시스템 연산

- 하나 이상의 CPU와 장치 제어기
 - 공유 메모리에 접근할 수 있도록 공통 버스로 연결되어 있다
- CPU와 장치는 병행 수행하기 때문에 메모리를 사용하기 위해 경쟁함





컴퓨터 시스템 연산

- 입출력 장치와 CPU는 병행 실행된다
- 각 장치 제어기는 특정 유형의 장치를 책임진다
- 각 장치 제어기는 지역 버퍼를 가진다
- CPU는 메인 메모리로 지역 버퍼로 혹은 지역 버퍼에서 메인 메모리로 데이터를 이동한다
- 입출력은 장치에서 제어기의 지역 버퍼로 데이터를 이동하는 것이다
- 장치 제어기는 **인터럽트**를 발생시켜 CPU에게 연산이 끝났음을 알린다



인터럽트의 공통 기능

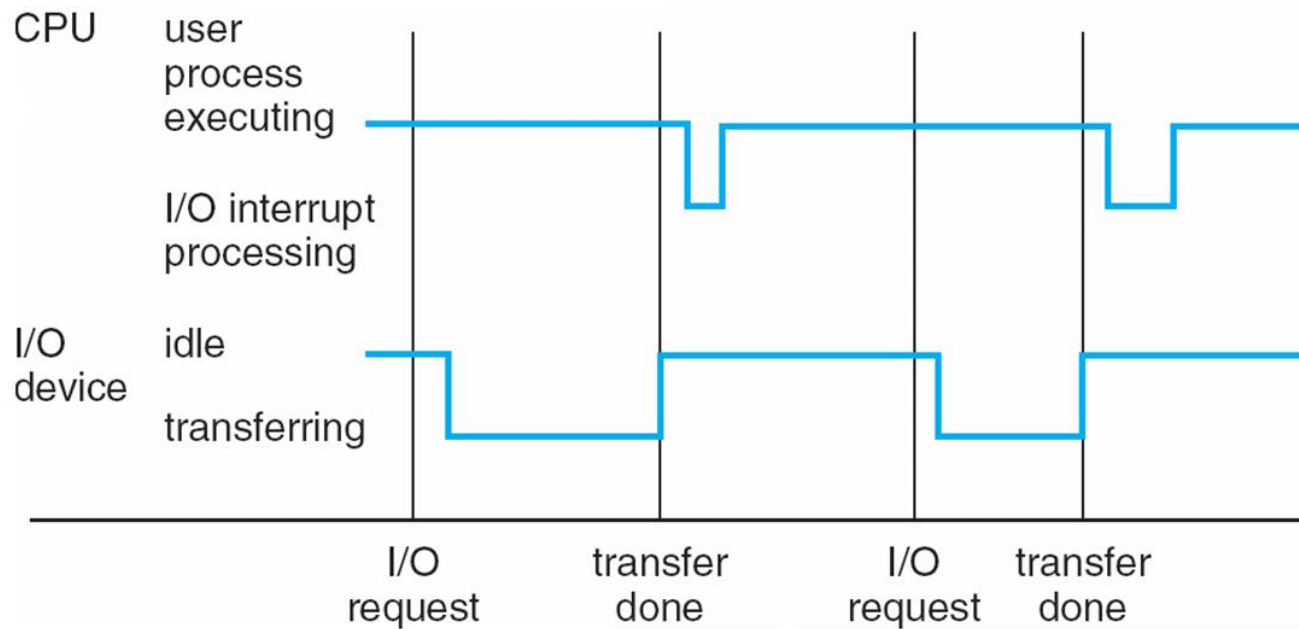
- 인터럽트는 인터럽트 서비스 루틴에게 제어권을 넘긴다.
 - 종류별로 서비스 루틴이 다를 수 있으며
 - 모든 서비스 루틴의 주소를 저장하고 있는 자료구조를 **인터럽트 벡터**라고 한다
- 인터럽트 아키텍처는 반드시 인터럽트 된 명령어의 주소를 저장해야 한다
- **트랩** 또는 **예외**는 소프트웨어가 발생시킨 인터럽트를 의미한다
 - 프로그램의 오류나 사용자 요청에 의해 발생된다
- 운영체제는 인터럽트가 발생해야 작동을 시작하는 방식(**interrupt driven**)



인터럽트 핸들링

- 운영체제가 레지스터와 프로그램 카운터를 저장하여 인터럽트 발생 순간의 CPU 상태를 보존한다
- 발생한 인터럽트의 유형을 파악한다
 - 폴링(polling)
 - 벡터를 이용한 시스템(vectored interrupt system)
- 담당 코드가 인터럽트 유형에 따른 대응 행동을 결정한다

인터럽트 처리 시간 흐름



입출력 구조

- 입출력이 시작된 후, 제어권은 입출력이 완료된 후에야 사용자 프로그램에게 넘어간다
 - Wait 명령어는 다음 인터럽트가 발생할 때까지 CPU를 아무 일도 안하게 만든다
 - 반복문을 돌며 기다린다(메모리 접근을 위한 경쟁을 유발한다)
 - 최대 하나의 입출력 요청만 대기할 수 있다
 - 2개 이상의 입출력을 동시에 처리할 수 없다
- 입출력이 시작된 후 제어권은 완료를 기다리지 않고 바로 사용자 프로그램에게 넘어간다
 - **시스템 호출(System call)** – 사용자가 입출력 완료를 대기할 수 있도록 운영체제에게 요청
 - **장치 상태 테이블(Device-status table)**은 장치마다 유형, 주소와 상태를 표시한다
 - OS는 장치의 상태를 결정하고 인터럽트를 추가하기 위하여 항목을 갱신할 때 테이블을 찾아본다

저장장치의 정의와 표기

- n 컴퓨터 저장장치의 기본단위는 비트이다. 한 비트는 0과 1 두 값 중 하나를 가진다.
- n 한 바이트는 8 비트이고 대부분의 컴퓨터에서는 가장 작은 편리한 저장 단위이다.
- n 워드는 그 컴퓨터 구조의 본연의 데이터 단위이다.
- n 1 킬로바이트 또는 KB는 1024 바이트
- n 1 메가바이트 또는 MB는 1,024바이트
- n 1 기가바이트 또는 GB는 1,024바이트
- n 1 테라바이트 또는 TB는 1,024바이트
- n 1 페타바이트 또는 PB는 1,024바이트
- n 컴퓨터 제조업체는 이들 숫자를 반올림하여 1 메가바이트를 백만 바이트, 1 기가바이트를 십억 바이트라 말한다
- n 네트워킹 수치는 이러한 일반 규칙에 대한 예외로 비트 단위로 표현된다(왜냐하면 네트워크는 데이터를 한 번에 1 비트씩 이동하기 때문이다).



저장장치 구조

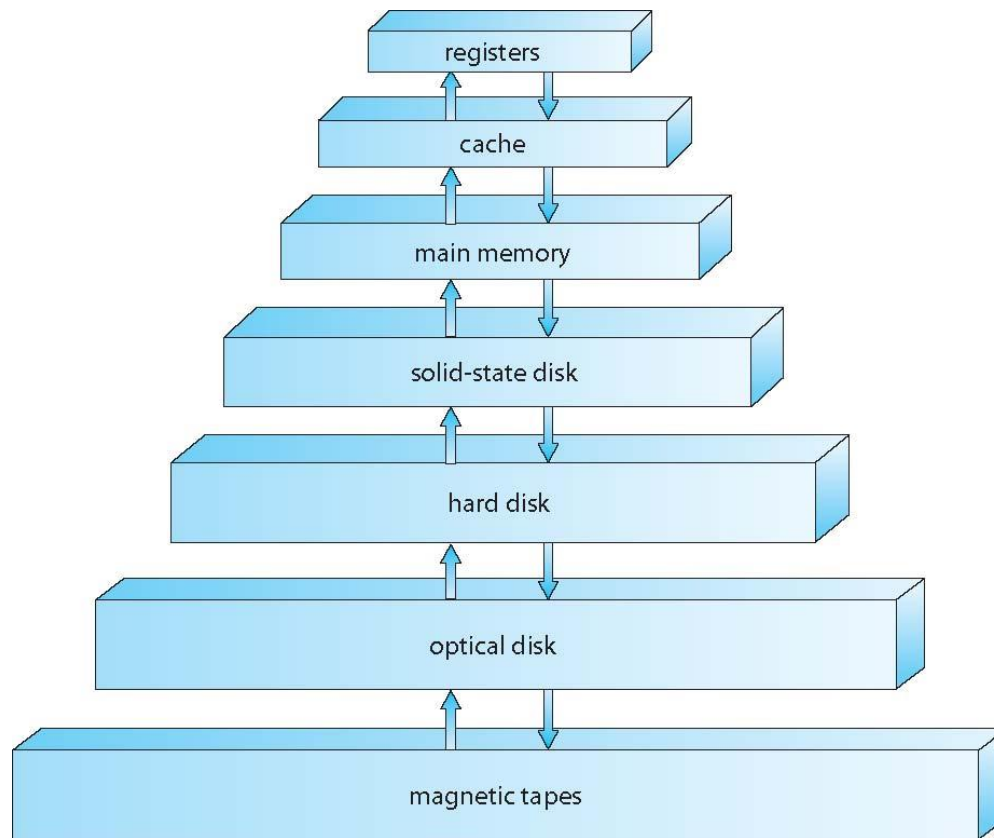
- 메인 메모리 – CPU가 직접 접근할 수 있는 유일한 대용량 저장 매체
 - 임의 접근(Random access)
 - 보통 휘발성(volatile)
- 보조 저장장치 – 대용량 비휘발성 저장 용량을 제공하는 메인 메모리의 확장
- 하드 디스크 – 자성 기록 가능 물질로 덮힌 단단한 금속 또는 유리 플래터
 - 디스크 표면이 논리적인 트랙으로 나뉘고 트랙은 다시 섹터로 나뉜다
 - 디스크 제어가 장치와 컴퓨터의 논리적인 상호 작용을 결정한다
- 반도체 디스크(Solid-state disks) – 하드 디스크 보다 빠른 비휘발성 저장 매체
 - 다양한 기술
 - 점점 더 대중화 되고 있음



저장장치 계층 구조

- 저장장치 시스템은 계층 구조를 형성함
 - 속도(Speed)
 - 비용(Cost)
 - 휘발성 여부(Volatility)
- 캐싱(Caching) – 더 빠른 저장장치로 정보를 복사하는 기법; 메인 메모리를 보조 저장장치의 캐시로 볼 수 있음
- 입출력을 관리하기 위해 장치마다 장치 드라이버(Device Driver)가 있음
 - 제어기와 커널 사이의 일관된 인터페이스를 제공

저장장치 계층 구조





캐싱

■ 중요한 원칙

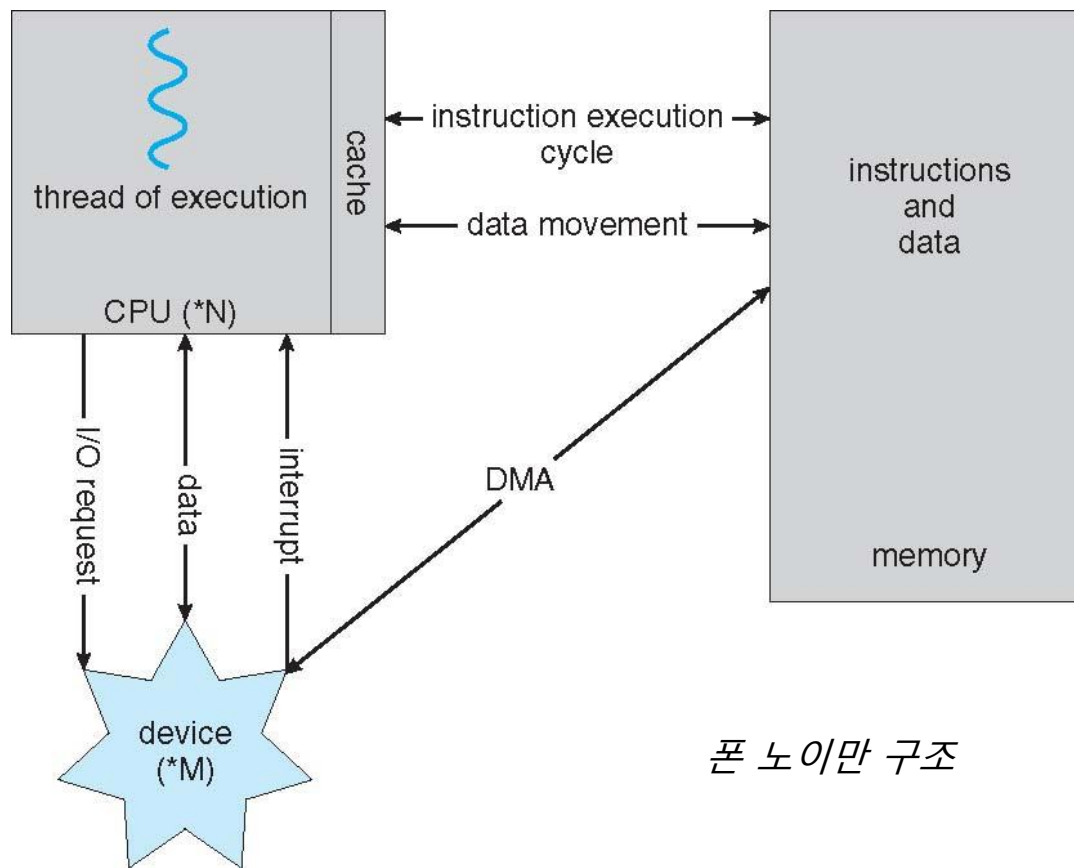
- 하드웨어, 운영체제, 소프트웨어 등 컴퓨터의 많은 레벨에서 수행된다
- 사용 중인 정보는 느린 장치에서 빠른 장치로 임시적으로 복사된다
- 빠른 저장장치(캐시)는 우선 원하는 정보가 이미 존재하고 있는지 점검한다
 - 존재하면, 캐시에서 직접 사용된다
 - 존재하지 않으면, 데이터를 캐시로 가져와서 캐시에서 사용된다
- 캐시는 데이터가 오는 소스 저장장치 보다 용량은 적다
 - 캐시 관리는 중요한 설계 사항
 - 캐시 용량과 교체 정책



직접 메모리 접근 구조

- 메인 메모리 속도에 근접한 속도로 정보를 전송할 수 있는 고속 입출력 장치를 위해 사용되는 기법
- 장치 제어기는 데이터 블록을 장치의 버퍼에서 메인 메모리로 CPU를 개입시키지 않고 직접 전송한다
- 바이트가 전송될 때마다 인터럽트를 발생시키지 않고 블록의 전송이 끝날 때마다 하나의 인터럽트만 발생시킨다

현대 컴퓨터가 작동하는 방식



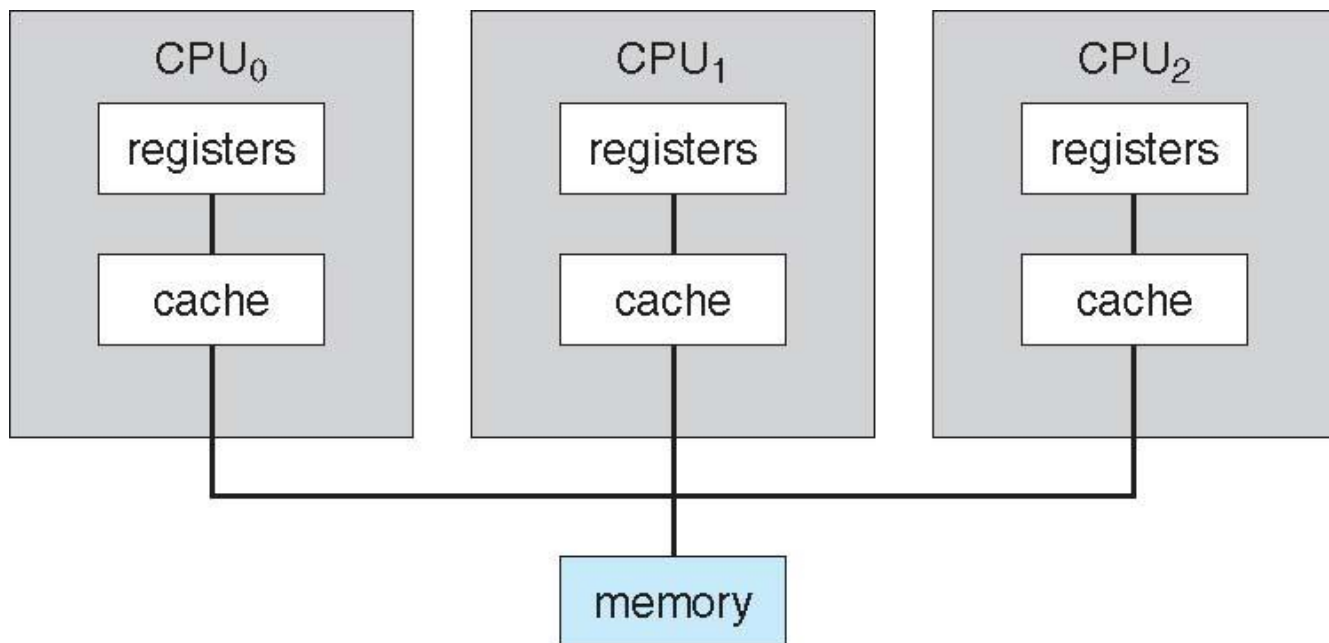
폰 노이만 구조



컴퓨터 시스템 구조

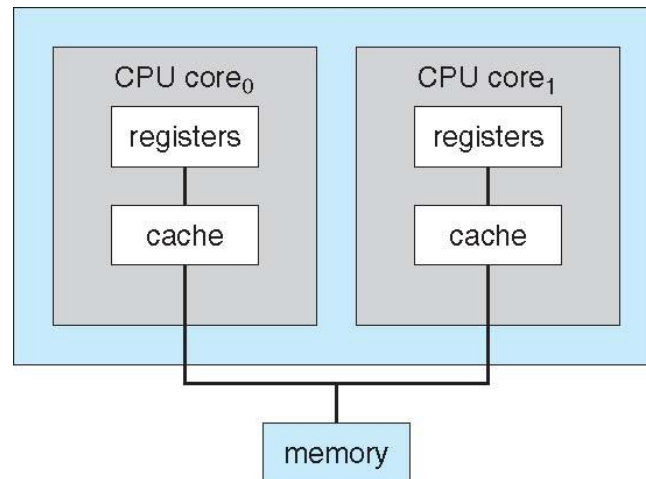
- 대부분의 시스템은 하나의 범용 CPU를 사용한다
 - 대부분의 시스템은 또한 전용 CPU도 가지고 있다
- **다중처리기(Multiprocessors)** 시스템의 사용과 중요성이 점점 증가하고 있음
 - 병렬 시스템, 밀결합 시스템이라고 불림
 - 장점
 1. 증가된 처리량(Increased throughput)
 2. 규모의 경제(Economy of scale)
 3. 증가된 신뢰성(Increased reliability) – 우아한 퇴보 또는 결함 허용
 - 두 가지 형태
 1. 비대칭 다중처리(Asymmetric Multiprocessing) – 각 처리기가 특별한 임무를 수행한다
 2. 대칭 다중처리(Symmetric Multiprocessing) – 각 처리기는 공통적인 임무를 수행한다

대칭 다중처리 구조



이중 코어 디자인

- 여러 칩과 다중코어(**multicore**)
- 시스템은 모든 칩을 포함하여 구성할 수도 있음
 - 개별 시스템을 새시 형태로 모아 구성하는 블레이드 서버도 있음

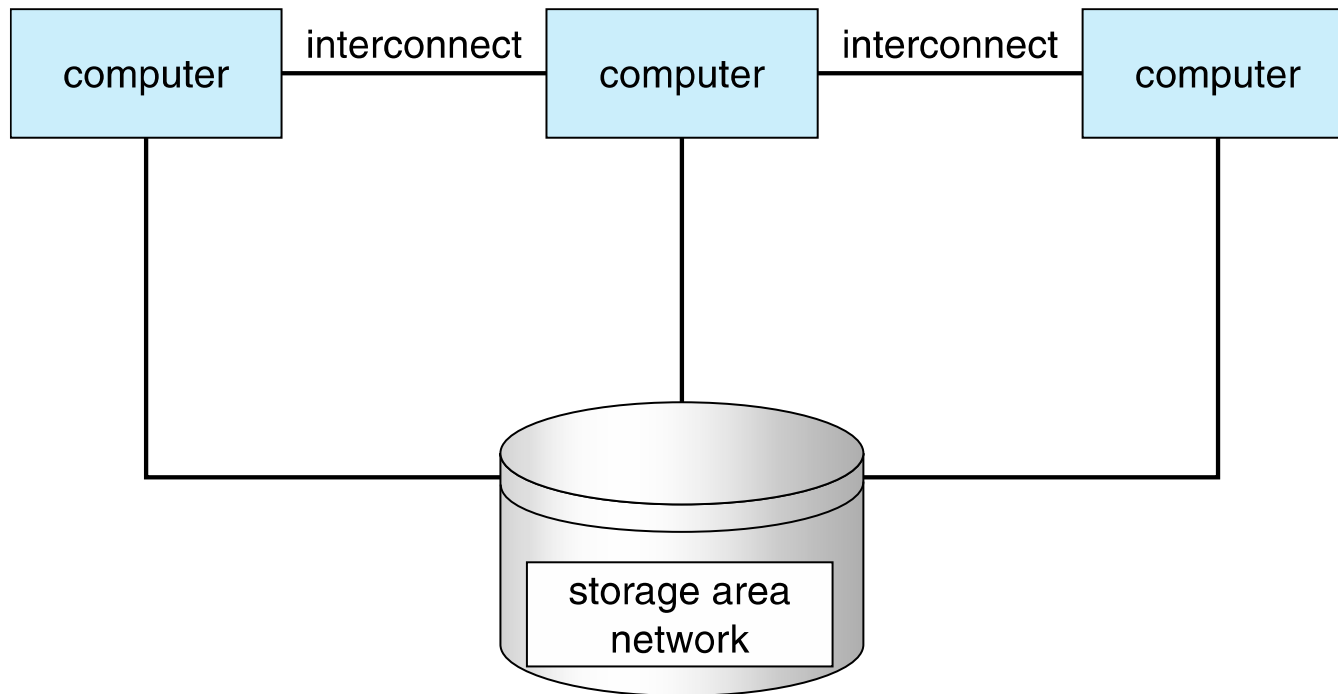




클러스터 시스템

- 다중처리 시스템과 같이 여러 개의 시스템도 같이 작업할 수 있다
 - 보통 **storage-area network (SAN)**를 통하여 저장장치를 공유
 - 실패해도 지속할 수 있는 **고가용(high-availability)** 서비스 제공
 - **비대칭 클러스터링(Asymmetric clustering)**에서는 기기 하나가 **긴급 대기(hot-standby)** 모드 상태
 - **대칭 클러스터링(Symmetric clustering)**에서는 여러 노드가 응용프로그램을 실행하고 서로를 모니터링함
 - 일부 클러스터는 고성능 컴퓨팅 환경(**high-performance computing**)을 제공
 - 응용프로그램은 병렬처리가 가능하도록 작성되어야 함
 - 일부 클러스터는 분산 락 관리자(**distributed lock manager**)를 제공하여 상충하는 연산을 피한다

클러스터 시스템





운영체제 구조

- 다중프로그래밍(일괄처리 시스템)은 효율이 중요
 - 한 명의 사용자가 CPU와 입출력 장치를 항상 작동하게 만들 수 없다
 - 다중프로그래밍은 CPU가 항상 하나의 작업을 수행할 수 있도록 여러 작업들을(코드와 데이터) 배치한다
 - 전체 작업의 부분집합이 메모리에 적재되어 유지된다
 - **작업 스케줄링(job scheduling)이 실행할 작업을 선택한다**
 - 실행 중인 작업이 무언가를 기다려야 한다면(예, 입출력) 운영체제는 다른 작업으로 전환한다



운영체제 구조

- 시분할 (Timesharing (multitasking))은 CPU가 빈번하게 다른 작업으로 전환하도록 논리적으로 확장한 것이다
- 따라서 사용자는 각 작업과 상호 작용 할 수 있고 대화식 컴퓨팅(interactive computing) 환경을 만들게 된다
 - 응답시간(Response time) 1초 미만이어야 한다
 - 각 사용자는 메모리에 실행 중인 프로그램을 적어도 하나는 가지고 있다 ⇒ 프로세스(process)
 - 동시에 여러 작업이 실행할 준비가 되어 있다면 ⇒ CPU scheduling
 - 프로세스를 전부 메모리에 적재할 수 없다면 스와핑(swapping)이 메모리에 들어거나 내보낸다
 - 가상메모리(Virtual memory)는 전체가 메모리에 적재되지 않은 프로세스의 실행을 가능하게 한다

다중 프로그래밍 시스템의 메모리 레이아웃

