1장. 서론 (Introduction)

순천향대학교 컴퓨터공학과
이 상 정

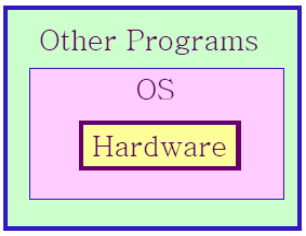
강의 목표 및 내용

- 목표
  - 운영체제의 주요 구성 요소 이해
  - 컴퓨터 시스템 구조에 대한 기본지식
- 내용
  - 운영체제가 하는 일
  - 컴퓨터 시스템 구성 및 구조
  - 운영체제의 구조 및 연산
  - 프로세스 관리, 메모리 관리, 저장 장치 관리
  - 보호와 보안, 분산 시스템, 전용 시스템, 계산 환경

운영체제(Operating System, OS)란?

❑ 운영체제란?

- 컴퓨터 하드웨어 관리하는 프로그램
- 컴퓨터 사용자와 컴퓨터 하드웨어 사이의 중재자 역할을 하는 프로그램



❑ 운영체제 목표

- 사용자 프로그램을 실행
- 컴퓨터 시스템을 사용자가 편리하고 용이하게 사용하여 문제 해결
- 컴퓨터 하드웨어를 효율적으로 관리

컴퓨터 시스템(Computer System) 구성 요소 (1)

❑ 컴퓨터 시스템의 4가지 구성 요소

- 하드웨어
  - 기본 계산용 자원(computing resource)을 제공
  - 중앙 처리 장치(CPU), 메모리, 입/출력(I/O) 장치
- 운영체제
  - 다양한 응용들과 사용자들 간 하드웨어 사용을 제어하고 중재
- 응용 프로그램
  - 사용자의 계산 문제를 해결하기 위해 시스템 자원(system resource)들이 어떻게 사용될 것인지를 정의
  - 워드 프로세서, 컴파일러, 웹 브라우저, 데이터베이스 시스템, 비디오 게임 등
- 사용자
  - 사람, 기계, 다른 컴퓨터 들

컴퓨터 시스템 구성 요소 (2)

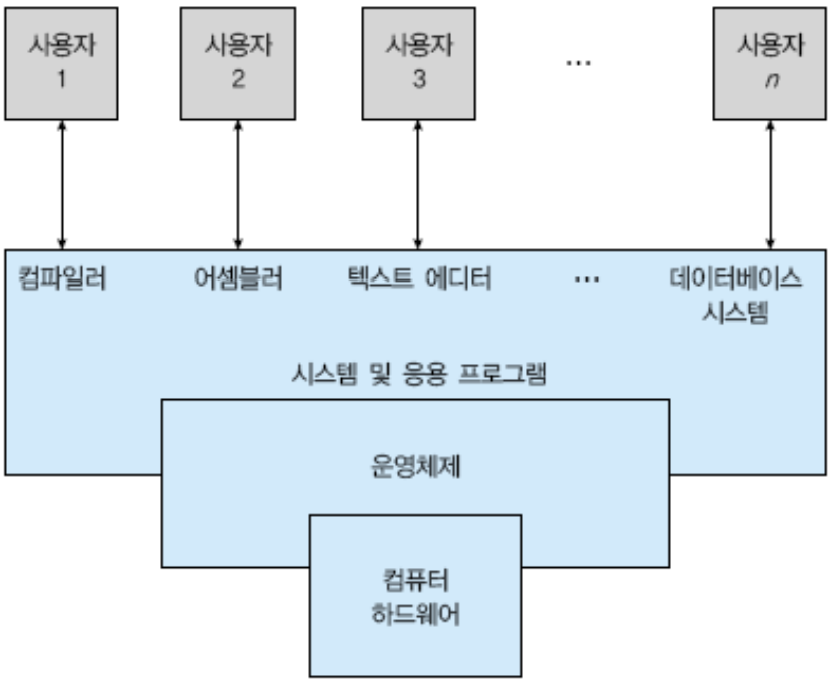


그림 1.1 컴퓨터 시스템 구성 요소에 대한 개략적 구성도

운영체제의 정의

- 운영체제는 자원 할당기 (resource allocator)
  - 모든 자원들을 관리하고 스케쥴(schedule)
  - 서로 상충되는 요청들을 조정 하여 효율적이고 공정하게 자원을 사용
- 운영체제는 제어 프로그램 (control program)
  - 컴퓨터 부적절한 사용과 에러를 방지하도록 프로그램들의 실행을 제어
- 운영체제는 커널 (kernel)
  - 컴퓨터 상에서 항상 실행되는 프로그램
    - 항상 메모리에 상주 (memory resident)
  - 다른 프로그램들(시스템 프로그램 또는 응용 프로그램)은 운영체제에 종속

컴퓨터 구동 (Startup)

- 부트스트랩 프로그램 (bootstrap program)
  - 전원이 켜지거나 재부트(reboot)될 때 실행되는 초기 프로그램
  - ROM이나 EEPROM에 저장
    - 펌웨어(firmware)라고 함.
  - 시스템의 모든 사항을 초기화
    - CPU 레지스터, 장치 제어기, 메모리 내용 등
  - 운영체제의 커널을 찾아 메모리에 적재하고 실행을 시작

컴퓨터 시스템 구조(Organization)

- 컴퓨터 시스템 연산(operation)
  - 공유 메모리에 접근 가능한 공통 버스에 연결된 하나 이상의 CPU와 여러 개의 장치 제어기(device controller)들로 구성
  - CPU와 장치 제어기는 메모리 사이클을 얻기 위해 경쟁하면서 병행 수행(concurrent execution)

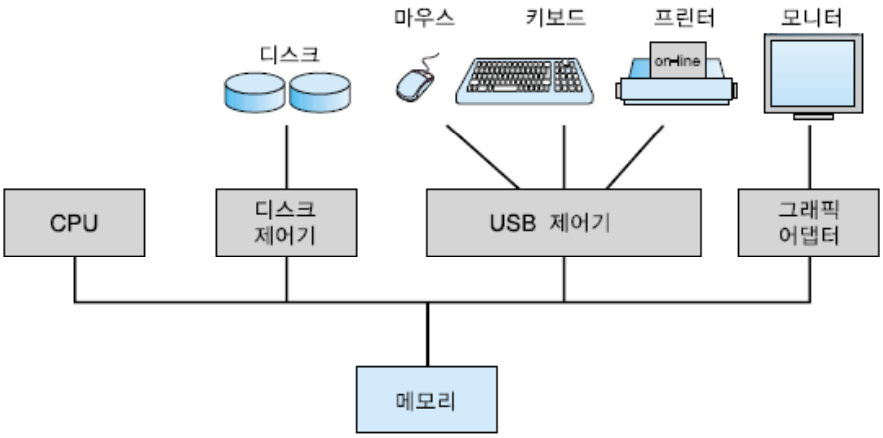


그림 1.2 현대의 컴퓨터 시스템

입/출력 연산

- 입/출력 장치들과 CPU는 병행 수행
- 입/출력 장치 제어기(I/O device controller)
  - 특정 장치 타입에 대해 동작
  - 지역 버퍼(local buffer)를 가짐 (data register)
- CPU는 주 메모리(main memory)와 지역 버퍼들 사이에 데이터 송수신을 제어
- 입/출력은 장치와 제어기의 지역 버퍼 사이 수행
- 장치 제어기는 입출력 연산 종료 시 인터럽트(interrupt)을 사용하여 CPU에 알림

인터럽트 (Interrupt) 소개

- CPU에 인터럽트 신호(signal)가 들어오면 실행 중인 작업을 멈추고 고정된 위치의 인터럽트 서비스 루틴(interrupt service routine)으로 이동하여 실행
  - 인터럽트 벡터(interrupt vector)가 인터럽트 서비스 루틴들의 시작 주소에 관한 정보 제공
  - 실행 중인 작업 주소를 저장하여 서비스 루틴 종료 후 복귀
- 다른 인터럽트 처리 중에 새로이 요청되는 인터럽트는 승인 거부 될 수 있음
  - 인터럽트 우선순위에 따라 승인 또는 거부 결정
- 트랩(trap)은 에러나 사용자 요청에 기인한 소프트웨어가 생성한 인터럽트
- 운영체제는 인터럽트 기반으로 구동

인터럽트 처리 (Interrupt Handling)

- 운영체제는 레지스터와 프로그램 카운터(PC)를 저장하여 실행 중인 CPU 상태를 보존
- 발생된 인터럽트의 형태를 파악:
  - 폴링 (polling)
  - 벡터 인터럽트 시스템
- 인터럽트의 형태에 따라 서로 다른 코드에 의해 적용되는 동작을 결정

입/출력 인터럽트 시간 일정 (Time Line)

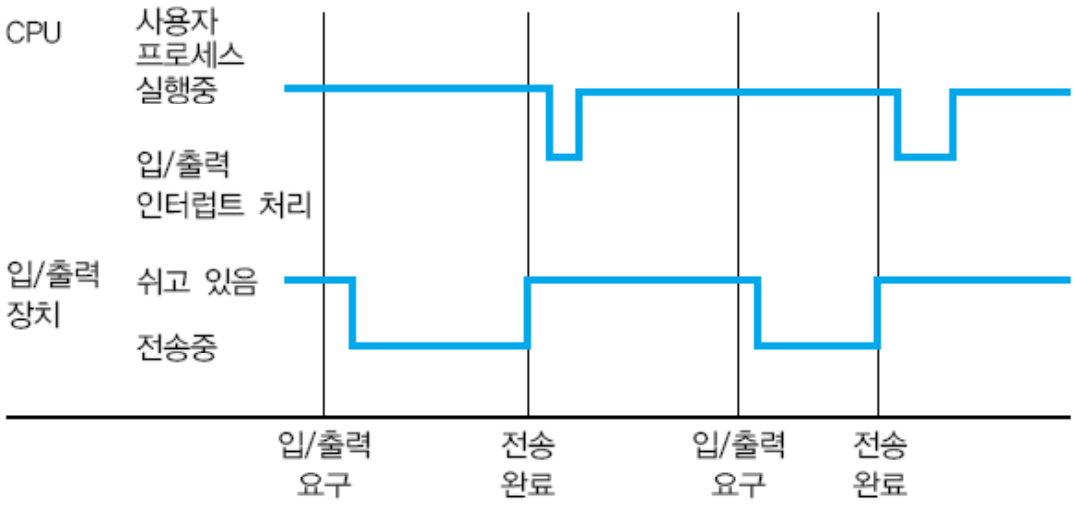


그림 1.3 출력을 수행하고 있는 단일 프로세스에 대한 인터럽트 시간 일정

저장 장치 구조 (Storage Structure)

- 주 메모리 (main memory)
  - CPU가 직접 접근 가능한 유일한 대량의 저장 장치
  - 주로 DRAM으로 구현
- 보조 저장 장치 (secondary storage)
  - 대용량의 비휘발성(nonvolatile) 저장장치로 주 메모리를 확장
  - 자기 디스크가 보편화
    - 디스크의 표면은 논리적으로 트랙(track)과 섹터(sector)로 분할
    - 디스크 제어기는 장치와 컴퓨터 간에 논리적인 상호 작용을 제어

저장 장치 계층 (Hierarchy)

- 저장 장치는 계층(hierarchy)으로 구성
  - 속도, 비용, 휘발성
- 캐싱 (caching)
  - 더 빠른 저장 장치로 정보를 복사
  - 주 메모리는 보조 저장 장치를 캐싱

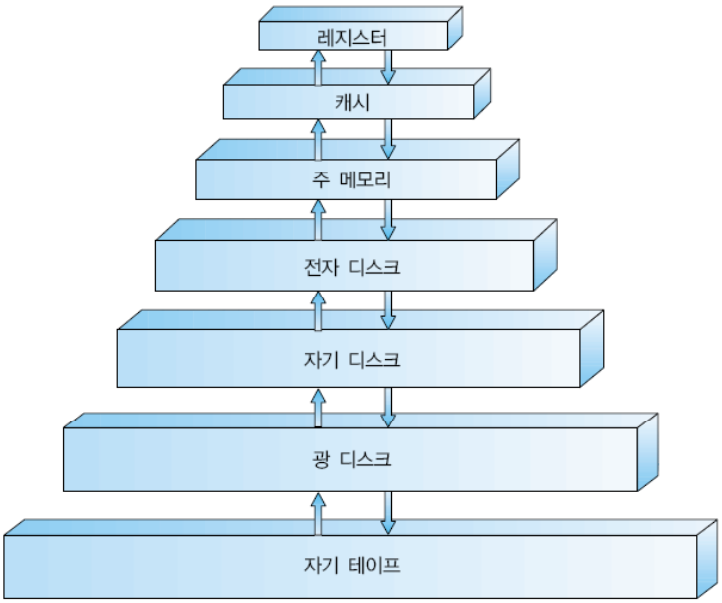


그림 1.4 저장 장치 계층

저장 장치 성능

수준	1	2	3	4
명칭	레지스터	캐시	주 메모리	디스크 저장 장치
전형적인 크기	1 KB	16 KB	16 GB	100 GB
구현 기술	다중 포트를 가지는 고유 메모리, CMOS	온칩 또는 오프칩 CMOS RAM	CMOS DRAM	자기 디스크
접근 시간(ms)	0.25 ~ 0.5	0.5 ~ 25	80~250	5,000,000
대역폭(MB/sec)	20,000~ 100,000	5000 ~ 10,000	1000 ~ 5000	20 ~ 150
관리자	컴파일러	하드웨어	운영체제	운영체제
백업 여부	캐시	주 메모리	디스크	CD 또는 테이프

그림 1.9 저장 장치 각 단계의 성능

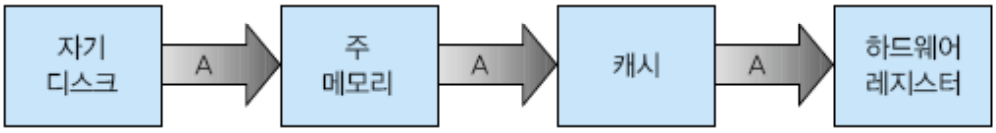


그림 1.10 정수 A를 디스크로부터 레지스터로 이동하는 과정

입/출력 구조 (I/O Structure)

- 운영체제는 각 장치 제어기마다 장치 드라이버(device driver)를 내장
  - 장치 드라이버는 장치 제어기의 동작을 이해하고, 입출력 장치에 대한 일관된 인터페이스 제공
- 입출력 연산
  1. 장치 드라이버는 장치 제어기의 레지스터들에 필요한 값을 적재
  2. 장치 제어기는 입출력 동작(예를들면 "키보드로 부터 한 문자 입력 ")을 위해 이들 레지스터의 내용을 조사
  3. 제어기는 장치로부터 자신의 지역 버퍼로 데이터 전송을 시작
  4. 데이터의 전송이 완료되면, 장치 제어기는 자신이 연산을 완료했음을 인터럽트를 이용하여 장치 드라이버에게 통보
  5. 장치 드라이버는 제어를 운영체제에게 반환하고 이 때 입력 완료인 경우에는 데이터 또는 데이터에 대한 포인터를 같이 반환
     - 다른 동작에 대해서는 장치 드라이버는 상태 정보를 반환

직접 메모리 접근 (Direct Memory Access, DMA)

- 인터럽트 구동 방식의 입/출력은 디스크 입/출력과 같은 대량의 데이터를 전송하는 데에는 높은 오버헤드(overhead)를 초래

- 직접 메모리 접근(Direct Memory Access, DMA)
  - 고속 전송이 가능한 입/출력 장치에 사용
  - 장치 제어기는 CPU의 개입 없이메모리로부터 자신의 버퍼 장치로 또는 버퍼로부터 메모리로 데이터 블록 전체를 전송
  - 속도가 느린 장치처럼 한 바이트마다 인터럽트가 발생하는 것이 아니라 블록 전송이 완료될 때마다 인터럽트가 발생

컴퓨터 시스템 동작

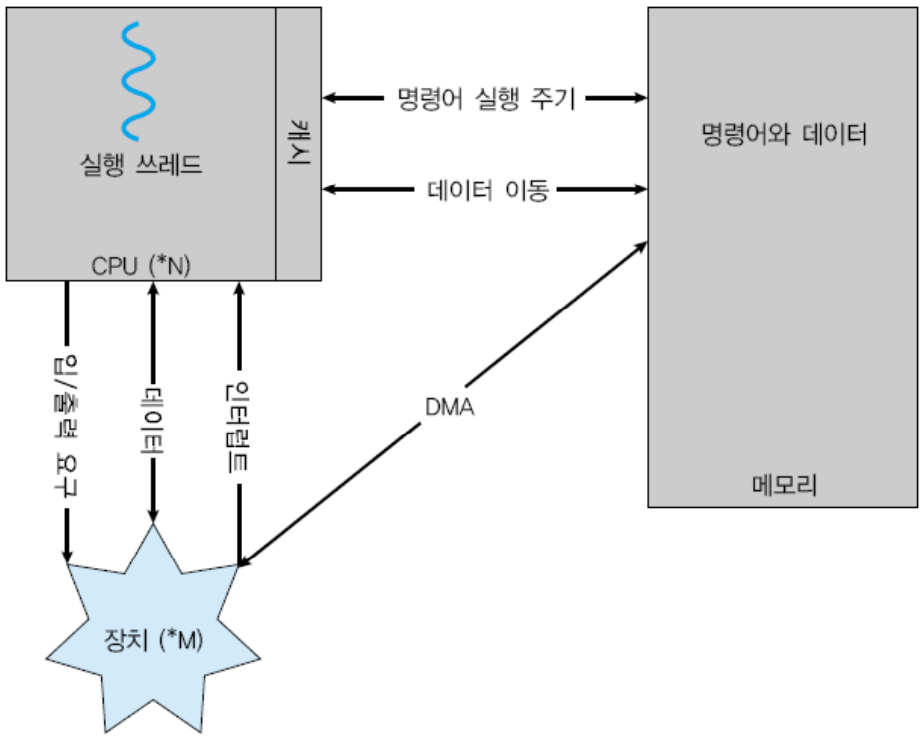


그림 1.5 현대 컴퓨터 시스템의 동작

컴퓨터 시스템 구조 (1)

- 단일 처리기 시스템(Single-Processor System)
  - 범용 처리기(processor, CPU)가 하나 있는 시스템
  - 디스크나 그래픽 제어기와 같은 특정 장치 처리기
- 다중 처리기 시스템(Multiprocessor System)
  - 다수의 처리기를 갖는 시스템
  - 대칭적 다중 처리(symmetric multiprocessing, SMP)
    - 각 처리기가 운영체제 내의 모든 작업을 동등하게 수행

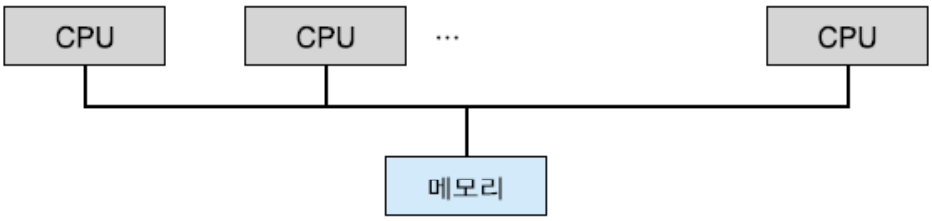


그림 1.6 대칭적 다중 처리 구조

컴퓨터 시스템 구조 (2)

- 비대칭적 다중 처리(asymmetric multiprocessing)
  - 하나의 주(master) 처리기가 시스템을 제어한다. 다른 처리기들은 주 처리기의 명령을 수행하거나 미리 정의된 태스크를 수행
- 다중 처리기 칩(chip multiprocessor), 멀티코아(multicore) 처리기
  - 하나의 칩에 여러 개의 코어(core)를 내장
- 클러스터형 시스템(clustered system)
  - 둘 이상의 독자적 시스템들을 연결하여 구성
  - 저장 장치를 공유하고 근거리 통신망(Local Area Network, LAN) 고속의 상호 연결망(interconnect)으로 연결

운영체제의 구조 (OS Structure)

❑ 다중 프로그래밍(multiprogramming)

- 단일 사용자는 CPU 또는 입/출력 장치를 항상 바쁘게 유지할 수 없음
- 다중프로그래밍(multiprogramming)은 CPU가 수행할 작업(코드와 데이터)을 항상 하나는 가지도록 작업(job)을 구성
- 운영체제는 한 번에 여러 작업을 메모리에 적재
- 운영체제는 메모리 내에 있는 작업 중에서 하나를 선택해 실행
- 선택된 작업이 입/출력의 종료를 기다리는 동안 운영체제는 다른 작업으로 전환해 수행

그림 1.7 다중 프로그래밍 시스템을 위한 메모리 배치도

시분할 (Time Sharing)

❑ 시분할(time sharing 또는 멀티태스킹: multi-tasking)은 다중 프로그래밍의 논리적 확장

- CPU가 다수의 작업들을 매우 빈번하게 교대가 일어나기 때문에 프로그램이 실행되는 동안 사용자들은 각자 자기의 프로그램과 상호 작용
- 응답 시간(response time)이 짧아야 하며, 통상 1초
- 각 사용자는 메모리에 최소한 하나의 프로그램을 가짐 => 프로세스(process)
- 여러 개의 작업이 동시에 실행준비가 되어 있으면, 시스템은 그들 중 하나를 선택 => CPU 스케쥴링
- 프로세스들이 모두 메모리에 놓일 수 없는 경우 스와핑(swapping)
  - 스와핑이란 필요에 의해 프로세스를 주 메모리에서 디스크로, 디스크에서 주 메모리로 옮기는 작업
- 가상 메모리(virtual memory)가 일부만이 메모리에 있는 작업의 수행을 허용

운영체제 연산

- 운영체제는 인터럽트 구동식(interrupt driven)
- 트랩(trap)(또는 예외 exception)은 오류 또는 사용자 프로그램의 운영체제 서비스 수행 요청에 의해 유발되는 소프트웨어에 의해 생성된 인터럽트
  - 0으로 나누기 또는 유효하지 않은 메모리 접근
- 운영체제와 사용자는 컴퓨터 시스템의 하드웨어와 소프트웨어 자원을 공유하기 때문에 사용자 프로그램의 오류가 현재 수행중인 프로그램에만 문제를 일으키도록 보장해야 함

이중 연산 모드

- 이중 연산 모드로 운영체제 코드의 실행과 사용자 정의 코드의 실행을 구분하여 운영체제를 보호
  - 사용자 모드(user mode), 커널 모드(수퍼바이저(supervisor) 모드, 시스템 모드, 특권 모드(privileged mode))
- 하드웨어가 모드 비트(mode bit) 제공
  - 커널 모드(0) 또는 사용자 모드(1)를 나타내는 비트
  - 시스템이 사용자 코드 또는 커널 코드 실행 여부를 구분
  - 일부 명령을 특권 명령(privileged instruction)으로 지정하여 커널 모드에서만 수행되도록 허용

시스템 호출(System Call)

- 시스템 호출(system call)은 사용자 프로그램이 자신을 대신하여 운영체제가 수행하도록 예약되어 있는 작업들을 운영체제에게 요청
  - 시스템 호출 시 커널 모드로 변경되고, 복귀 시 사용자 모드로 리셋

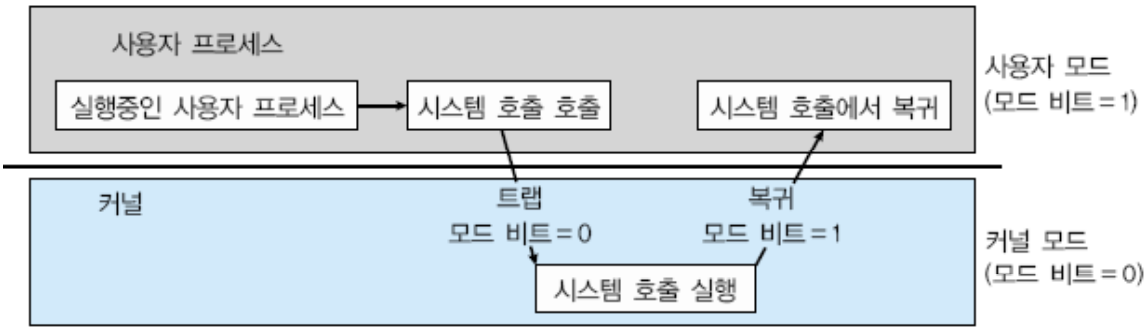


그림 1.8 사용자 모드에서 커널 모드로의 전환

프로세스 관리(Process Management)

- 프로세스는 실행 중인 프로그램
- 프로세스는 자신의 일을 수행하기 위해 CPU 시간, 메모리, 파일, 그리고 입/출력 장치를 포함한 여러 가지 자원이 필요
  - 단일 쓰레드(thread) 프로세스는 다음 수행할 명령을 지정하는 하나의 프로그램 계수기(program counter)를 가짐
  - 다중 쓰레드 프로세스는 복수개의 프로그램 계수기를 가짐
- 운영체제는 프로세스 관리와 연관해 다음과 같은 활동에 대한 책임
  - 사용자 프로세스와 시스템 프로세스의 생성과 제거
  - 프로세스의 일시 중지와 재수행
  - 프로세스 동기화(synchronization)를 위한 기법 제공
  - 프로세스 통신을 위한 기법 제공
  - 교착상태(deadlock) 처리를 위한 기법 제공

메모리 관리 (Memory Management)

- 주 메모리는 CPU와 입/출력 장치에 의하여 공유되는, 빠른 접근이 가능한 데이터의 저장소
- 실행되는 모든 명령들과 데이터는 메모리 내에 상주
  - 폰 노이만 방식 컴퓨터
- 운영체제는 메모리 관리와 관련하여 다음과 같은 일을 담당
  - 메모리의 어느 부분이 현재 사용되고 있으며 누구에 의해 사용되고 있는지를 추적
  - 어떤 프로세스(또는 그 일부)들을 메모리에 적재하고 제거할 것인가를 결정
  - 필요에 따라 메모리 공간을 할당(allocation)하고 회수해야 한다.

저장장치 관리 (Storage Management)

- 주 메모리는 모든 데이터와 프로그램을 수용하기에 용량이 너무 작고, 전원이 꺼질 경우 데이터가 소멸되므로 주 메모리 내용을 저장(backup)하기 위해 보조 저장 장치(secondary storage)가 필요
  - 대부분의 현대의 컴퓨터 시스템은 디스크(disk)를 프로그램과 데이터를 위한 주된 온라인 저장 매체로 사용
- 운영체제는 디스크 관리를 위하여 다음과 같은 기능을 담당
  - 자유 공간(free space)의 관리
  - 저장 장치 할당
  - 디스크 스케줄링

입/출력 시스템 관리 (I/O System Management)

- 입/출력 시스템은 다음과 같이 구성
  - 버퍼링(buffering): 전송 중 데이터를 임시 저장
  - 캐싱(caching): 데이터의 일부를 고속의 저장장치에 저장
  - 스풀링(spooling): 한 작업의 출력을 다른 작업의 입력과 중첩
  - 일반적인 장치 드라이버 인터페이스
  - 특정 하드웨어 장치들을 위한 드라이버

파일 시스템 관리 (File System Management)

- 파일(file)은 파일 생성자에 의해 정의된 관련 정보의 집합
  - 일반적으로 파일은 프로그램(소스와 목적 프로그램 형태)과 데이터를 바이트(byte)로 표현
  - 파일은 사용을 쉽게 하기 위해 통상 디렉토리(directory)들로 구성
- 운영체제는 파일 관리를 위하여 다음과 같은 일을 담당
  - 파일의 생성 및 제거
  - 디렉토리 생성 및 제거
  - 파일과 디렉토리를 조작하기 위한 프리미티브(primitive)의 제공
  - 파일을 보조 저장 장치로 사상(mapping)
  - 안정적인(비휘발성) 저장 매체에 파일을 백업

보호와 보안 (Protection and Security)

- 보호(protection)란 컴퓨터 시스템이 정의한 자원에 대해 프로그램, 프로세스, 또는 사용자들의 접근을 제어하는 기법
- 보안(security)은 컴퓨터 시스템을 외부 또는 내부의 공격(attack)을 방어하는 것이 바로 기능

- 보호와 보안을 제공하기 위해서는 시스템의 모든 사용자들을 구분
  - 사용자 이름과 연관된 사용자 식별자(user ID)의 리스트를 유지
  - 사용자 식별자는 사용자의 모든 프로세스나 쓰레드, 파일에 연관
  - 그룹 식별자의 리스트로 사용자의 집합을 구분
  - 특권(privilege)을 사용하여 제한된 장치를 접근

계산 환경 (Computing Environments)

- 전통적 계산(Traditional Computing)
  - 네트워크에 연결된 PC들과 파일과 프린트 서비스를 제공하는 서버들로 구성
- 클라이언트 서버 계산(Client-Server Computing)
  - 클라이언트에 의해 생성되는 요구를 서버가 서비스 응답
  - 계산-서버(compute-server)는 서비스를 요청하는 클라이언트에 인터페이스를 제공 (데이터베이스 서버)
  - 파일-서버(file-server)는 클라이언트가 파일을 생성, 갱신, 읽기 및 제거할 수 있는 파일 시스템 인터페이스를 제공 (웹 서버)

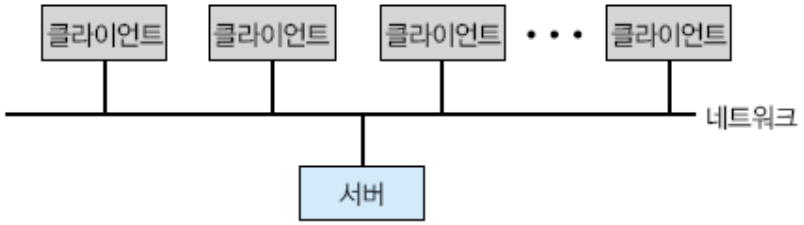


그림 1.11 클라이언트-서버 시스템의 일반 구조

피어간 계산(Peer-to-Peer Computing, P2P)

- P2P는 클라이언트와 서버가 서로 구별되지 않음
  - 모든 시스템 상의 모든 노드가 피어로 간주
  - 피어는 서비스를 요청하느냐 제공하느냐에 따라 클라이언트 및 서버로 동작
  - 노드는 먼저 피어간 네트워크에 참가(join)해야 함
    - 노드가 네트워크에 참가할 때 네트워크의 중앙 검색 서비스(central lookup service)에 자신이 제공하는 서비스를 등록
      또는
    - 서비스 발견 프로토콜(service discovery protocol)을 사용하여 네트워크 상의 모든 노드에게 서비스 요청 메시지를 방송(broadcast)
  - Napster나 Gnutella와 같은 파일 공유 서비스 등이 예

웹 기반 컴퓨팅(Web-Based Computing)

- 웹은 도처에 있으며(ubiquitous) 다양한 장치에 의한 접근 허용
  - PC가 아직 가장 널리 퍼진 접근 장치
  - 다양한 장치들이 네트워크화 되어 웹 접근을 허용
- 네트워크 연결을 비슷한 서버 풀 내에서 분배하는 적재 밸런서(load balancer)와 같은 새로운 부류의 장치가 탄생
- 웹 클라이언트로 동작했던 Windows 95와 같은 운영체제는 동시에 웹 서버와 웹 클라이언트로 동작할 수 있는 Windows ME와 Windows XP로 발전

과제: 연습문제

- 연습문제 1.2
- 연습문제 1.10
- 연습문제 1.11
- 연습문제 1.17

- 풀이는 PPT로 작성하고, 문제도 기술