

본 강의에서 수업자료로 이용되는 저작물은
저작권법 제25조 수업목적 저작물 이용 보상금제도에 의거,
한국복제전송저작권협회와 약정을 체결하고 적법하게 이용하고 있습니다.
약정범위를 초과하는 사용은 저작권법에 저촉될 수 있으므로
수업자료의 재 복제, 대중 공개·공유 및 수업 목적 외의 사용을 금지합니다.

2023. 3 . 02 .

부천대학교·한국복제전송저작권협회

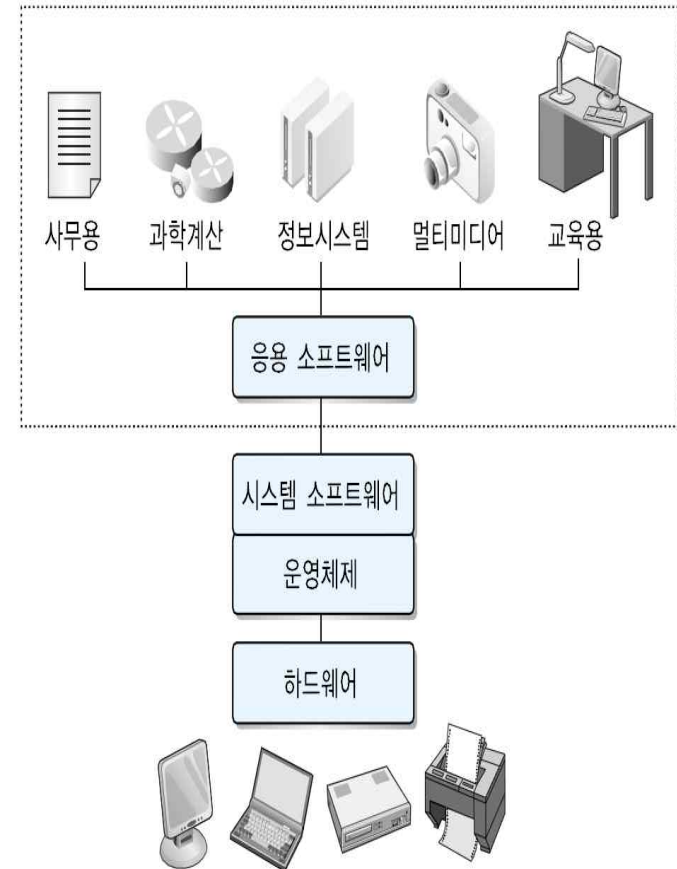
운 영 체 제

학습 내용

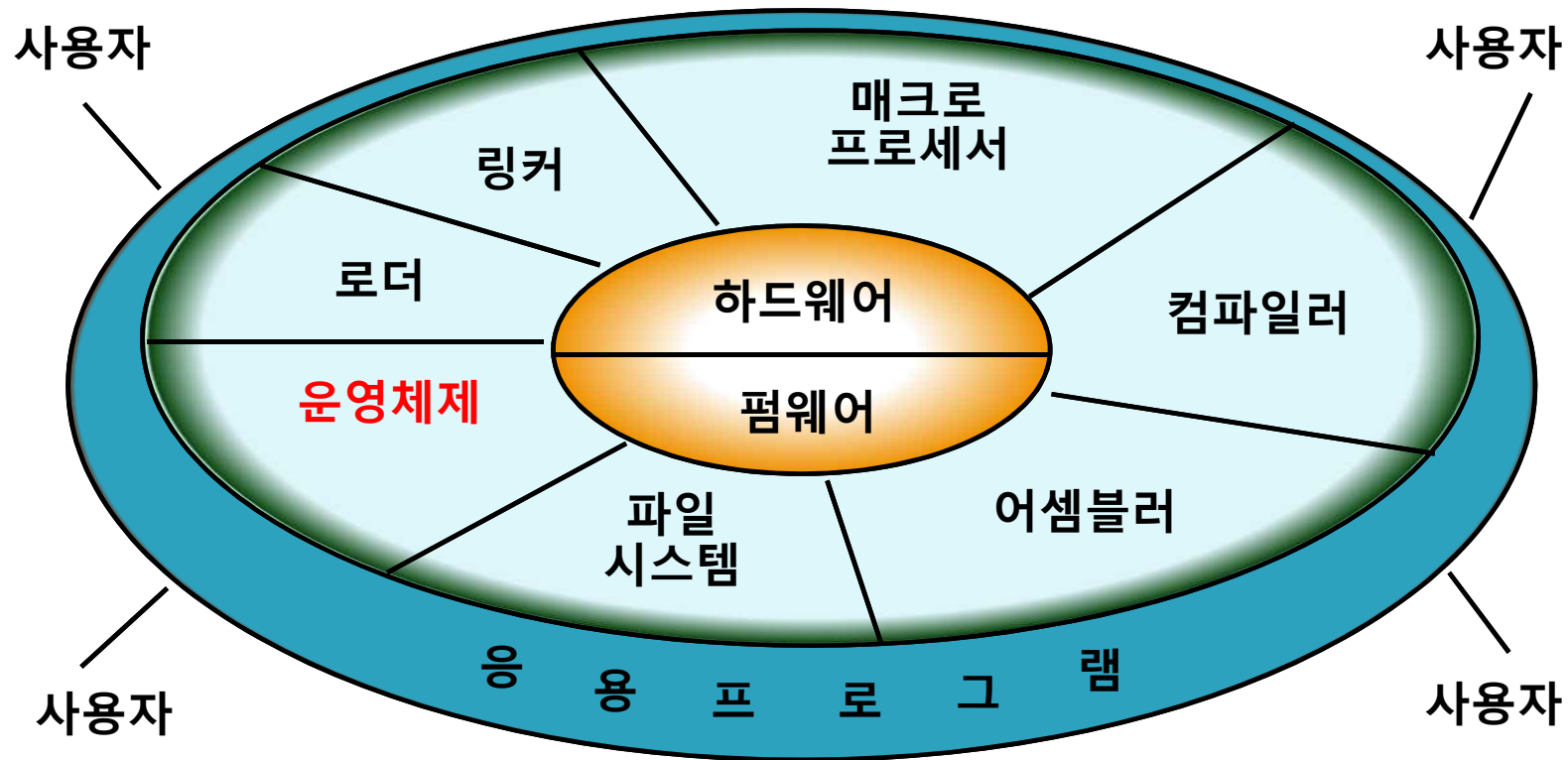
- ❖ 1장 운영체제 개요
 - 시스템 소프트웨어의 종류
 - 운영체제의 역할 및 목적
 - 운영체제의 발달
 - 운영체제의 분류
 - 운영체제의 유형별 특징
 - 운영체제의 기능
 - 운영체제의 구성
 - 운영체제의 자원 관리

소프트웨어

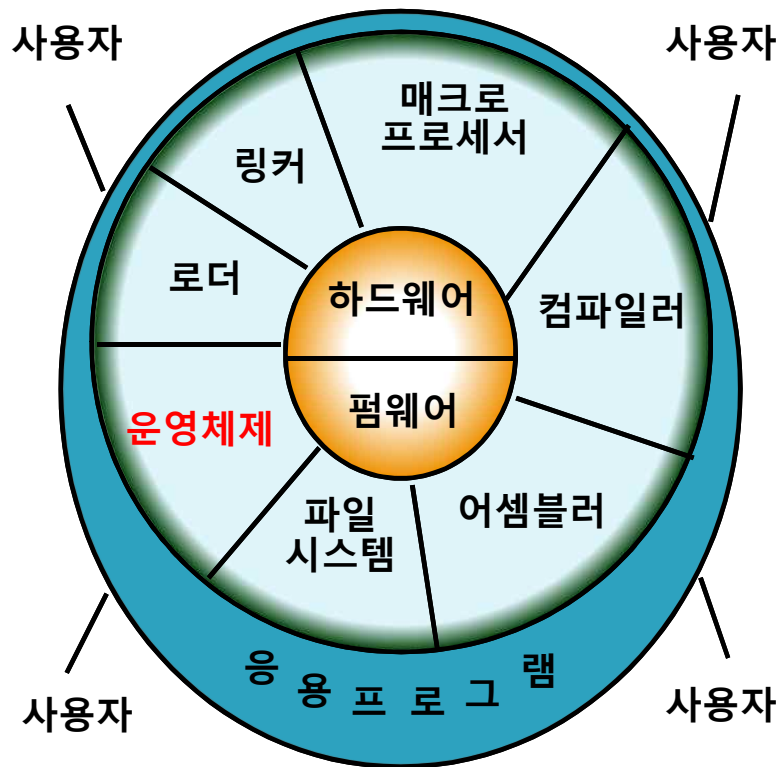
- 소프트웨어
 - 명령들이 모여서 하나의 프로그램을 형성하며, 프로그램들이 모여서 집합을 형성한 것이 소프트웨어
- 시스템 소프트웨어
 - 컴퓨터 시스템을 제어하고 운영하는 프로그램
 - 운영체제(DOS, UNIX, Window 7, Window 10)
 - 컴파일러(C, java 컴파일러 등)
 - 어셈블러
 - 링커
 - 로더
 - 입출력 제어 프로그램 등
- 응용 소프트웨어
 - 시스템 소프트웨어를 기반으로 특정한 응용 분야에서 특수 목적을 위해 사용할 수 있는 프로그램
 - OA 관련 소프트웨어(엑셀, 파워포인트 등)
 - 그래픽 관련 소프트웨어



시스템 소프트웨어의 계층구조(1 / 2)



시스템 소프트웨어의 계층구조(2/2)

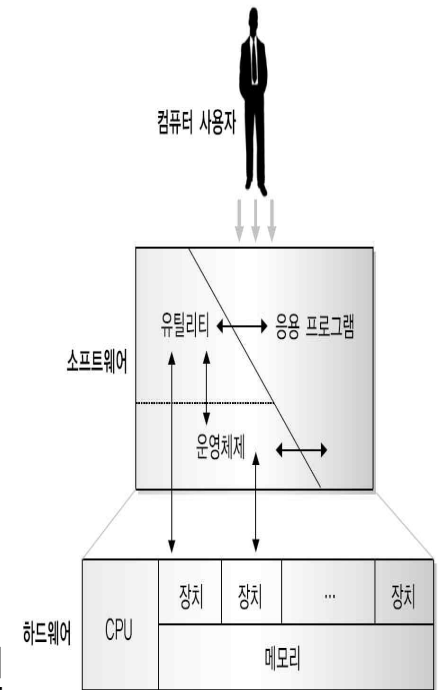


컴퓨터 시스템의 자원들을 효율적으로 관리하며, 사용자가 컴퓨터를 편리하고 효과적으로 사용할 수 있도록 환경을 제공하는 여러 프로그램의 모임으로, 제어 프로그램과 처리 프로그램으로 구분

- 제어 프로그램 : 시스템 전체의 작동 상태 감시, 작업의 순서 지정, 작업에 사용되는 데이터 관리 등의 역할 수행
 - 감시 프로그램(Supervisor Program)
 - 작업 제어 프로그램(Job Control Program)
 - 자료 관리 프로그램(Data Management Program)
- 처리 프로그램 : 제어 프로그램의 지시를 받아 사용자가 요구한 문제를 처리하기 위한 프로그램
 - 언어 번역 프로그램(Language Translator Program)
 - 서비스 프로그램(Service Program)
 - 문제 프로그램(Problem Program)
- 운영체제의 종류 : Windows, UNIX, LINUX, MS-DOS 등

운영체제의 개요

- 컴퓨터 하드웨어와 컴퓨터 사용자 간의 매개체 역할을 하는 시스템 소프트웨어
- 프로그램의 실행을 제어하는 소프트웨어
 - 자원의 할당, 스케줄링, 입출력 제어, 데이터관리 등의 서비스를 제공
- 주요 기능(처리 내용 기준)
 - File Management – 입출력 및 데이터 관리
 - Job Management - 자원할당(memory,CPU,..), Job 스케줄링
 - => Job(작업) : job이란 사용자가 컴퓨터에 요청한(부여한) 일의 단위
 - Task Management - TASK의 처리 상태를 감시하고 효율적으로 처리
 - => Task(업무) : 컴퓨터의 처리 단위인 TASK로 분할하여 Job을 실행
- 작업 수행하는 기준
 - => 일괄,다중 프로그래밍,시분할 처리, 다중 처리, 병행 처리, 분산 처리로 발전

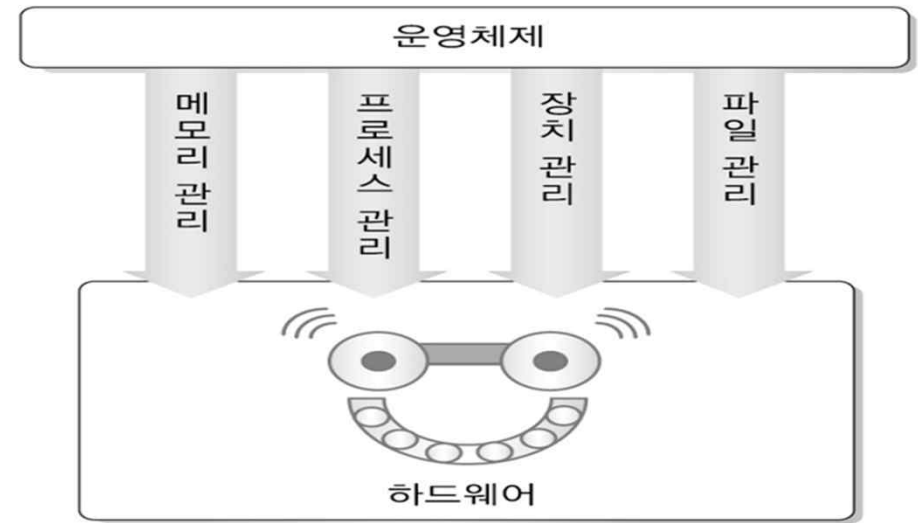
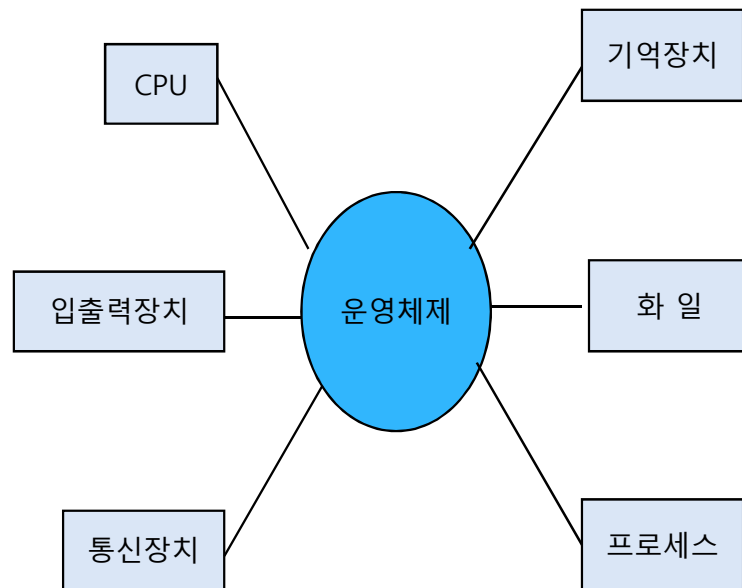


컴퓨터 시스템 구성요소와 운영체제

운영체제의 역할

- 운영체제 역할

- 운영체제는 컴퓨터 하드웨어와 응용 프로그램 간의 인터페이스 역할을 하며 **CPU, 메모리와 같은 컴퓨터 자원을 관리하고 사용자에게 편의를 제공**

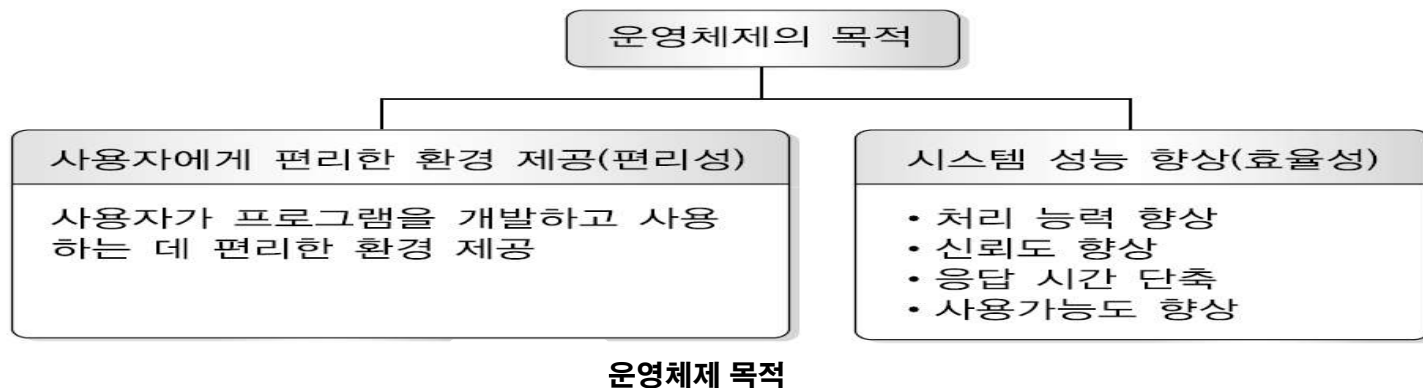


운영체제의 자원 관리 기능

운영체제의 목적 (1/4)

■ 운영체제의 목적

- **편리성** : 사용자가 프로그램을 개발하고 사용하는 데 좀 더 편리한 환경 제공.
ex: 개인용 컴퓨터(PC, Personal Computer)의 GUI 환경
- **효율성** : 자원을 효과적으로 사용하기 위해 각 프로그램을 유기적으로 결합하여 시스템 전체 **성능 향상**
=>성능을 향상시키는 방향으로 설계
- * **시스템 성능 평가 기준**
 - 처리 능력 (Throughput): 시스템의 생산성을 나타내는 대표 지표로 단위 시간당 처리하는 작업량.
 - 신뢰도 (Reliability): 하드웨어(펌웨어), 소프트웨어가 실패 없이 주어진 기능을 수행할 수 있는 능력.
 - 응답 시간(Turn around time) : 사용자가 시스템에 작업을 의뢰한 후 반응을 얻을 때까지의 시간.
(시분할 방식 시스템과 온라인 시스템에서 사용하는 용어, 일괄 처리 시스템에서는 Turn Around Time)
 - 사용가능도(가동률Availability) : 사용자가 일정 기간 동안 컴퓨터를 실제로 사용한 시간(비율)



운영체제의 목적 (2/4)

1. 사용자의 편의성 제고

* 컴퓨터 하드웨어의 복잡한 구성을 알지 못하는 사용자에게, 손쉽게 컴퓨터를 이용할 수 있도록 하는 방향으로 운영체제가 구현됨

*사례

- 키보드 입력과 문자 출력 형태
- 마우스 입력, 2차원 그래픽을 이용한 아이콘 입력
- 윈도우 위주의 GUI(Graphic User Interface)
- 펜 입력, 음성 입출력, 화상(정지 및 동영상), 애니메이션

운영체제의 목적 (3/4)

2. 시스템 성능의 극대화 (효율성)

(1) 처리능력(Throughput)의 증대

“일정 시간 내에 시스템이 처리하는 일의 양”

- ▶ 시스템 전체의 일 처리 능력을 강조.
- ▶ 사람의 개입을 최소화하는 방향으로 설계.
- ▶ 작업의 자동화된 연속처리가 가능하도록 설계
- ▶ CPU 대기시간을 짧게 : 다중 프로그래밍 도입

(2) 응답시간(Turnaround Time) 단축

“일의 처리를 컴퓨터에 의뢰하고 나서 결과를 얻을 때까지 걸리는 시간”

- ▶ 연속처리 기법(개념), ON-Line, 실시간 처리 개념 도입
- ▶ 다중 프로그래밍 개념의 도입, 시분할 처리 개념 도입

운영체제의 목적(4/4)

2. 시스템 성능의 극대화 (효율성)

(3) 사용 가능도(Availability) 증대

“컴퓨터시스템 내의 한정된 자원을 신속하고 충분히 지원해 줄 수 있는 정도”

- ▶ 자원의 양과 관계가 있음.
- ▶ 각종 자원에 대한 정보 관리가 필요하며, 신속히 지원해 줄 수 있는 스케줄링 기법 필요
- ▶ 시스템 유지보수, 오동작에 대한 자동적 대처 기법도 도입

(4) 신뢰도(Reliability) 향상

“시스템이 정확하게 작동되는 정도”

- ▶ 하드웨어 오류의 자체 회복 시도 기능
- ▶ 소프트웨어적 오류에 대한 상세한 메시지
- ▶ 하드웨어적 시스템의 신뢰성 향상 구축
 - Dual, Duplex 시스템

운영체제의 발달(1/4)

1. 운영체제 발달 요인

- ▶ 하드웨어의 비약적 발전
- ▶ 주기억 공간의 대용량화(새로운 기억 소자)
- ▶ 대용량, 고속의 자기 디스크 출현 : 가상기억 공간 개념
- ▶ 프로세서(CPU)의 처리 속도 증가
- ▶ 컴퓨터 사용자의 서비스 요구
- ▶ 시스템 운영자의 서비스 요구
- ▶ 컴퓨터 프로그래밍 언어의 발전
 - 운영체제의 구현 언어의 발전

2. 컴퓨터 출현 초기 (1940-50년대 초기)

: NO-OS : 초기 시스템

- ▶ 사용자가 컴퓨터 하드웨어를 직접 조작
- ▶ 입력장치 : 카드 리더
- ▶ 실행 상황은 표시 램프로서 확인, 수동으로 조작
- ▶ 작업(Job) 실행을 위한 여러 단계를 수작업으로 진행하기 때문에 시스템의 유휴 상태가 길어짐
- ▶ 컴퓨터 사용 예약시간 예측 불가
- ▶ 정상 실행을 위한 준비 시간이 길어짐

운영체제의 발달(2/4)

3. 상주 모니터(Resident Monitor) 개념 출현

▶ 출현 원인

- NO-OS의 문제점인 작업 준비 단계와 실행 단계에서 수작업으로 처리되는 부분을 프로그램(OS)하여 주기억 공간에 상주시킴
- 컴퓨터 조작용은 프로그래머의 요청에 따라서 작업을 진행시킴. 이에 따른 유휴 시간의 방지를 위하여 다른 작업으로 자동적으로 진행될 수 있도록 주기억 공간에 상주하는 프로그램의 필요성 대두됨
- 컴퓨터 프로그래밍 언어의 발전
 - 운영체제의 구현 언어의 발전

▶ 상주 모니터(Resident Monitor)

- 전문적인 조작용이 하던 일련의 작업 내용을 프로그램으로 작성하여 주기억 공간 내에 상주시킴.
- 프로그래머가 작성한 프로그램 처리에 대한 각종 지시 사항은 작업 제어 카드 (JCL)로 만들고, 이들을 해석하여 자동 처리가 되도록 함.
- 작업 제어 카드의 예 : 전문적인 조작용이 하던 일련의 작업 내용을 프로그램으로 작성하여 주기억 공간내에 상주시킴.
- 상주 모니터의 구성 예 : 제어 카드 해석기, 장치 구동기, 작업 순서화 루틴, 인터럽트 처리 루틴 등.

운영체제의 발달(3/4)

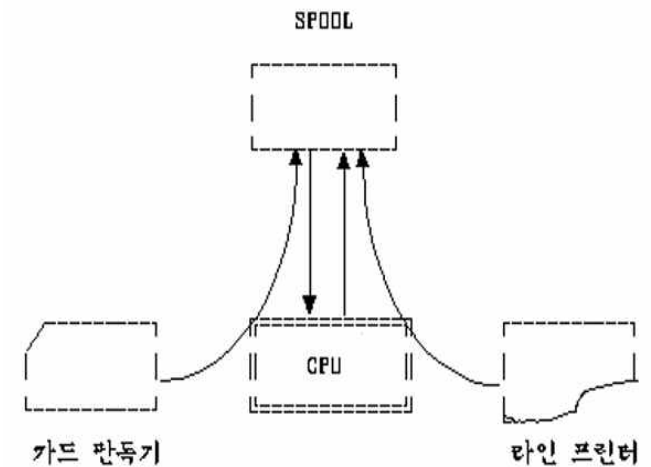
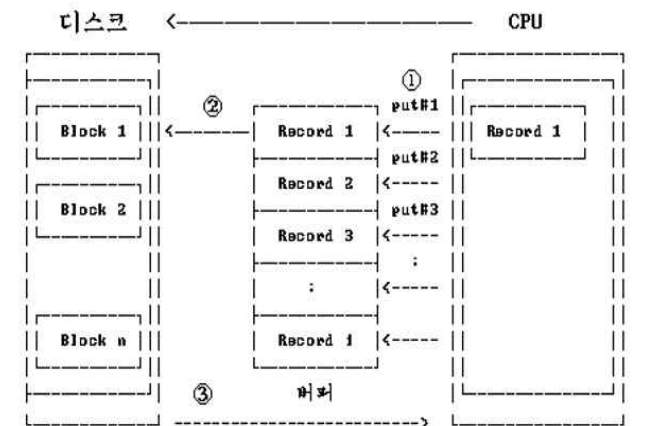
4. 버퍼링과 스푼링 개념의 도입

▶ 버퍼링(Buffering)

- 출현 이유 : 입출력 장치와 CPU 간의 처리 속도 차이에서 발생하는 CPU의 대기(유휴) 시간을 최소화하기 위해 필요하며, CPU의 효율적인 시간 관리를 지향.
- 주기억 장치의 일부를 버퍼로 설정함.
- 실제 입출력은 블록(Block) 단위로 버퍼와 I/O 기기 사이에서 처리됨.
- 느린 I/O 기기로 인한 CPU의 대기시간을 최소화하며, CPU의 효율적 관리가 가능해짐.

▶ 스푼링 : SPOOLING

- 출현 이유 : 시스템이 보유하고 있는 유한한 개수의 입출력 장치의 사용을 원활하게 하며, 입출력 장치의 사용을 원하는 사용자에게 지원이 용이하게 함
- 자기 디스크의 일부를 [스풀] 영역으로 사용함
- 주기억 장치와 입출력 장치를 직접 연결하지 않고, 그 사이에[스풀]을 위치시킴.



운영체제의 발달(4/4)

5. 다중 프로그래밍 개념의 도입

▶ 출현 이유

- single user 시스템인 경우, 자원의 낭비가 심해진다.
- 특히, 저속의 입출력 장치와 고속 CPU인 경우, I/O를 하는 동안에는 CPU가 쉬게 되는 문제점이 발생함

▶ 방안

- 하나의 작업이 I/O를 하는 동안, 다른 작업이 CPU를 쓸 수 있도록 여러 개의 프로그램을 주기억 공간에 적재시킴
=> 다중 프로그래밍 (Multiprogramming)

▶ 여러 개의 프로그램을 적재할 수 있는 충분한 기억공간

▶ 중앙 처리 장치 (CPU) 스케줄링 기법 및 CPU 보호 문제

▶ 입출력 작업을 독립적으로 수행할 수 있는 방안 강구

(I/O 프로세서 또는 Channel 개념 등 도입)

▶ 교착 상태(deadlock) 처리 방법

▶ 병행 제어 (Concurrent Control)를 위한 기법

▶ 인터럽트(Interrupt) 개념 도입 등

운영체제의 분류

- 동시 사용자 수에 따라
 - Single-user system
 - Multi-user system
- 작업 처리 방법에 따라
 - Single-tasking system
 - Multi-tasking system
(Multiprogramming system)
- 사용 환경에 따라
 - Batch processing system
 - Time-sharing system
 - Distributed processing system
 - Real-time system

운영체제의 유형

- 운영체제 운영 기법(작업 수행하는 방식에 따라)
 - 일괄 처리 시스템(Batch Processing System)
 - 다중 프로그래밍 시스템(Multi-Programming System)
 - 시분할 시스템(Time Sharing System)
 - 다중 처리 시스템(Multi-Processing System)
 - 실시간 처리 시스템(Real Time Processing System)
 - 다중 모드 처리 시스템(Multi-Mode Processing System)
 - 분산 처리 시스템(Distributed Processing System)

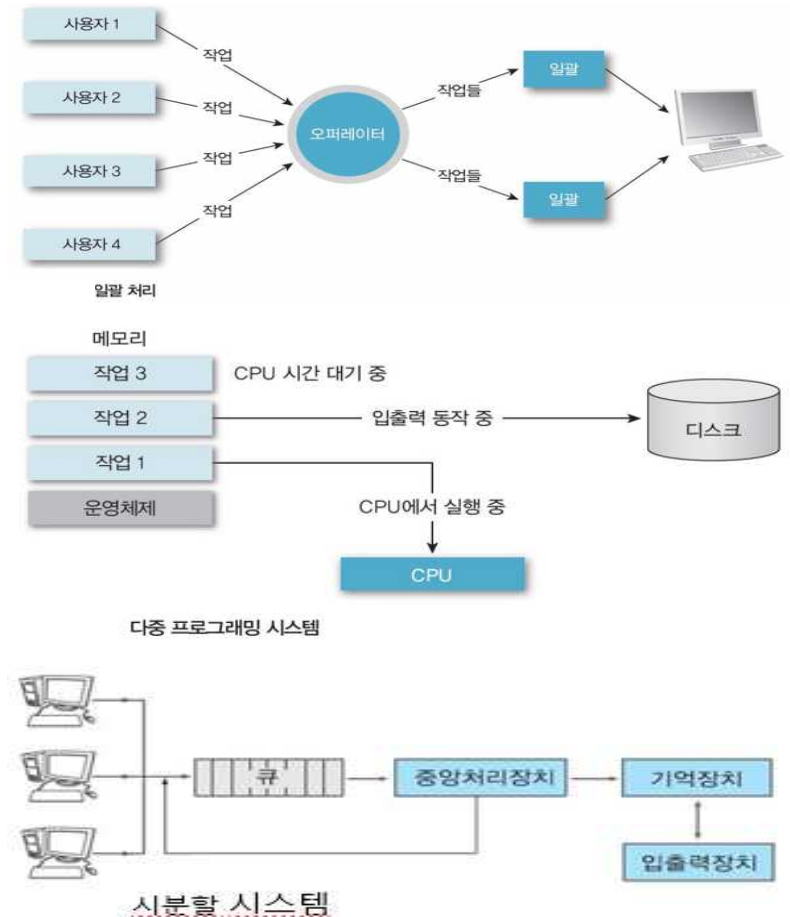
발달 과정:

일괄 처리	-> 다중 프로그래밍, 다중 처리, 시분할, 실시간 처리	-> 다중 모드	-> 분산처리시스템
1세대	2세대	3세대	4세대
(1950년대)	(1960년대)	(1960년대 중반 ~ 1970년대 중반)	(1970년대 중반 ~ 1990년대)

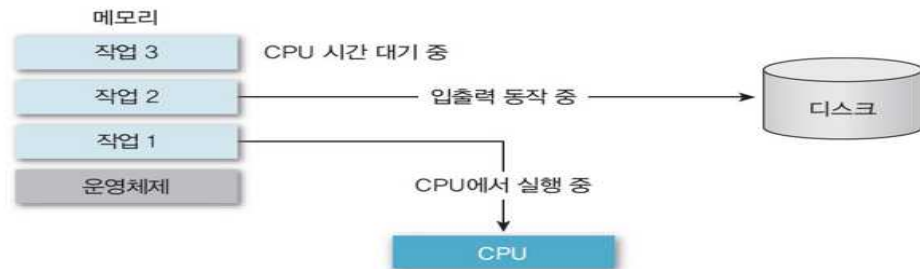
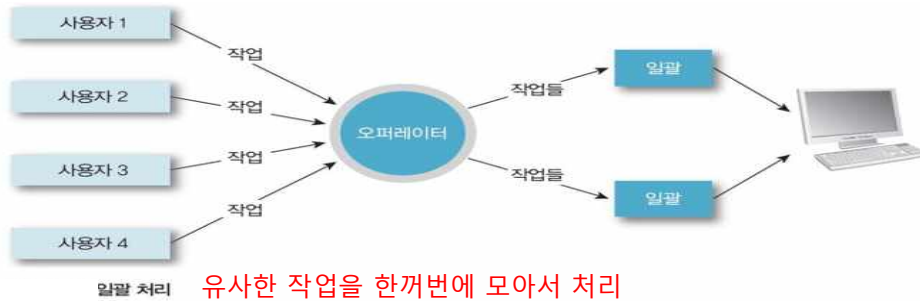
운영체제의 유형별 특징(1/3)

운영체제 운용 기법(작업을 수행하는 방식)

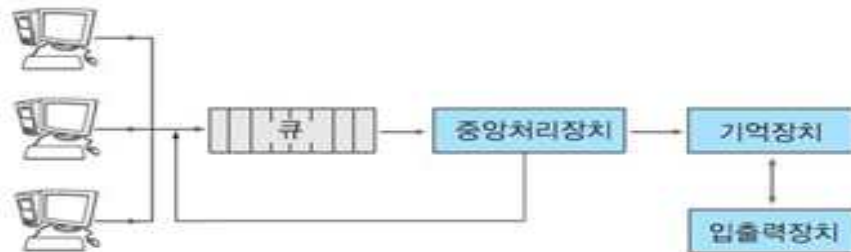
일괄 처리 (Batch Processing) 시스템	<ul style="list-style-type: none"> • 초기의 컴퓨터 시스템에서 사용된 형태로, 일정량 또는 일정 기간 동안 데이터를 모아서 한꺼번에 처리하는 방식 • 컴퓨터 시스템을 효율적으로 사용할 수 있음 • 사용자 측면에서는 반환(응답) 시간이 늦지만 하나의 작업이 모든 자원을 독점하므로 CPU 유휴 시간이 줄어듦 • 급여 계산, 지불 계산, 연말 결산 등의 업무에 사용
다중 프로그래밍 (Multi-Programming) 시스템	<ul style="list-style-type: none"> • 하나의 CPU와 주기억장치를 이용하여 여러 개의 프로그램을 동시에 처리하는 방식 • 하나의 주기억장치에 2개 이상의 프로그램을 기억시켜 놓고, 하나의 CPU와 대화하면서 동시에 처리함
시분할 (Time Sharing) 시스템	<ul style="list-style-type: none"> • 여러 명의 사용자가 사용하는 시스템에서 컴퓨터가 사용자들의 프로그램을 번갈아 가며 처리해 줌으로써 각 사용자에게 독립된 컴퓨터를 사용하는 느낌을 주는 것이며 라운드 로빈(Round Robin) 방식이라고도 함 • 여러 사용자가 각자의 단말 장치를 통하여 동시에 운영체제와 대화하면서 각자의 프로그램을 실행함 • 하나의 CPU는 같은 시점에서 여러 개의 작업을 동시에 수행할 수 없기 때문에, CPU의 전체 사용 시간을 작은 작업 시간량(Time Slice)으로 나누어서 그 시간량 동안만 번갈아 가면서 CPU를 할당하여 각 작업을 처리함 • 다중 프로그래밍 방식과 결합하여 모든 작업이 동시에 진행되는 것처럼 대화식 처리가 가능함



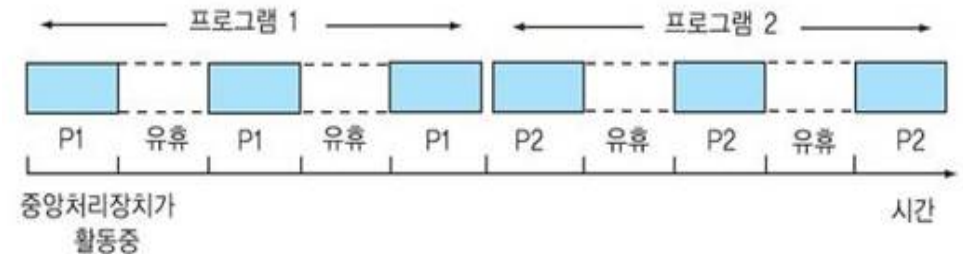
운영체제의 유형별 특징(2/3)



다중 프로그래밍 시스템 입출력 작업일 때 메인 메모리 활용도 증가 시키는 처리 (일괄처리의 단점 보완)



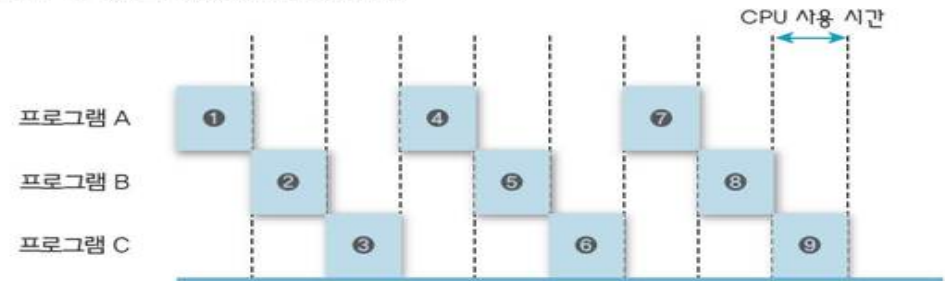
시분할 시스템 사용자 대기 시간을 줄여 다수의 사용자가 동시에 컴퓨터 자원을 공유하도록 하는 방식(일괄처리의 단점 보완)



순차성 실행



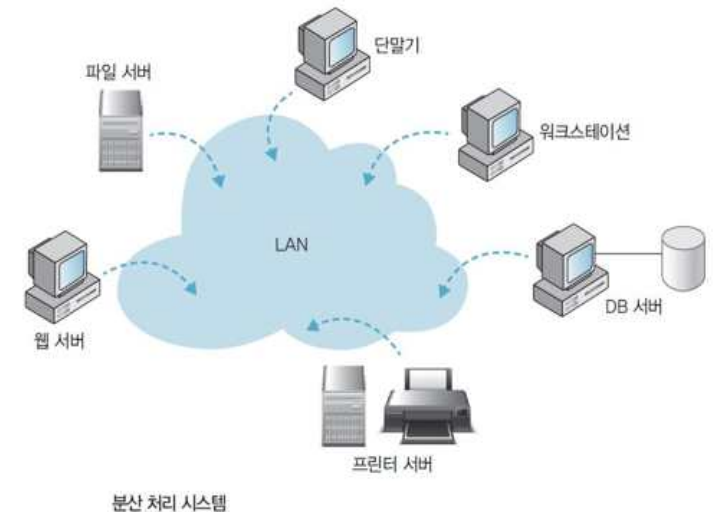
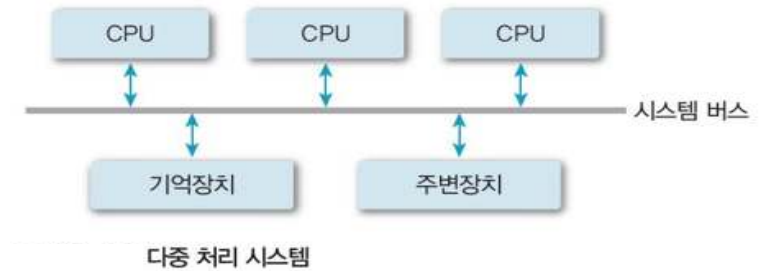
다중 프로그래밍 시스템의 처리 방법 예



시분할 시스템의 처리 방법 예

운영체제의 유형별 특징(3/3)

다중 처리 (Multi-Processing) 시스템	<ul style="list-style-type: none"> 여러 개의 CPU와 하나의 주기억장치를 이용하여 여러 개의 프로그램을 동시에 처리하는 방식 하나의 CPU가 고장나더라도 다른 CPU를 이용하여 업무를 처리할 수 있으므로 시스템의 신뢰성과 안정성이 높음
실시간 처리 (Real Time Processing) 시스템	<ul style="list-style-type: none"> 데이터 발생 즉시, 또는 데이터 처리 요구가 있는 즉시 처리하여 결과를 산출하는 방식 우주선 운행이나 레이더 추적기, 핵물리학 실험 및 데이터 수집, 전화교환장치의 제어, 은행의 온라인 업무, 좌석 예약 업무, 인공위성, 군함 등의 제어 업무 등 시간에 제한을 두고 수행되어야 하는 작업에 사용됨
다중 모드 처리(Multi-Mode Processing)	일괄 처리 시스템, 시분할 시스템, 다중 처리 시스템, 실시간 처리 시스템을 한 시스템에서 모두 제공하는 방식
분산 처리(Distributed Processing) 시스템	<ul style="list-style-type: none"> 여러 개의 컴퓨터(프로세서)를 통신 회선으로 연결하여 하나의 작업을 처리하는 방식 각 단말 장치나 컴퓨터 시스템은 고유의 운영체제와 CPU, 메모리를 가지고 있음



발달 과정

일괄 처리 시스템 → 다중 프로그래밍, 다중 처리, 시분할, 실시간 처리 시스템 → 다중 모드 → 분산 처리 시스템

1 세대 ->

2 세대

-> 3 세대 -> 4 세대

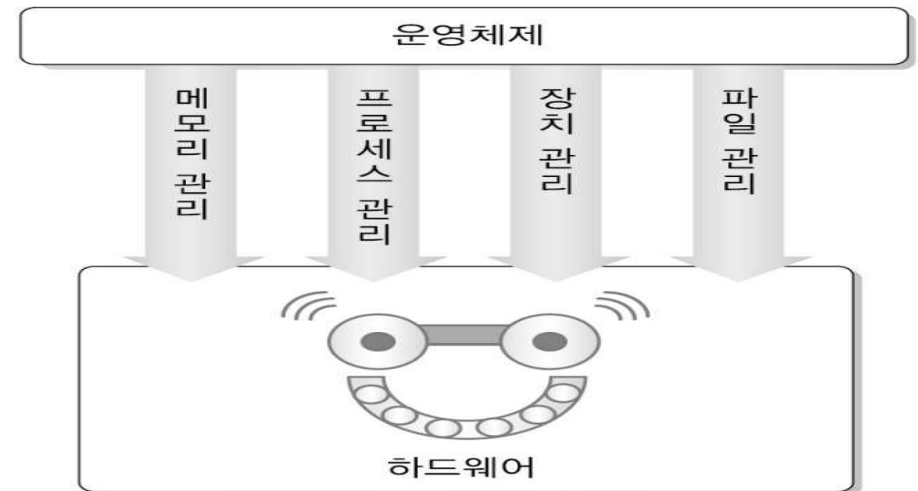
운영체제의 기능(1/2)

- 운영체제의 기능

- 운영체제는 프로그램을 실행하는 데 필요한 환경과 자원을 제공하고 관리하기 위한 크고 복잡한 시스템
- 논리적으로 작은 모듈로 구성되며 각 부분은 명확하게 정의
- 대부분의 시스템은 자원 관리와 프로그램을 위한 인터페이스 역할을 수행

- 자원 관리 기능

- 대부분 메모리, 프로세스, 장치, 파일 등의 시스템
- 구성 요소를 제공하며 이를 자원이라 하며,
- 운영체제는 이런 자원을 관리하는 역할을 수행.



운영체제의 자원 관리 기능

운영체제의 기능(2/2)

■ 자원 관리 기능

■ 프로세스 관리

- 프로세스를 주제로 프로세스의 상태와 변환 관련 기술과 제어, 스레드, 병행 프로세스(상호배제 및 동기화), 교착 상태, 프로세스 스케줄링의 기능

■ 메모리 관리

- 현재 메모리의 어느 부분이 사용되고, 누가 사용하는 지를 점검하는 기능
- 기억 공간에 어떤 프로세스를 저장할 지를 결정하는 기능
- 기억 공간을 할당하고 회수하는 방법 결정하는 기능

■ 보조기억장치 관리

- 메인 메모리의 공간이 제한적이므로 컴퓨터 시스템은 보조기억장치를 이용해 메인 메모리의 내용을 저장
- 디스크 관리를 위해 비어 있는 공간 관리, 저장 장소 할당, 디스크 스케줄링 기능

■ 장치 관리

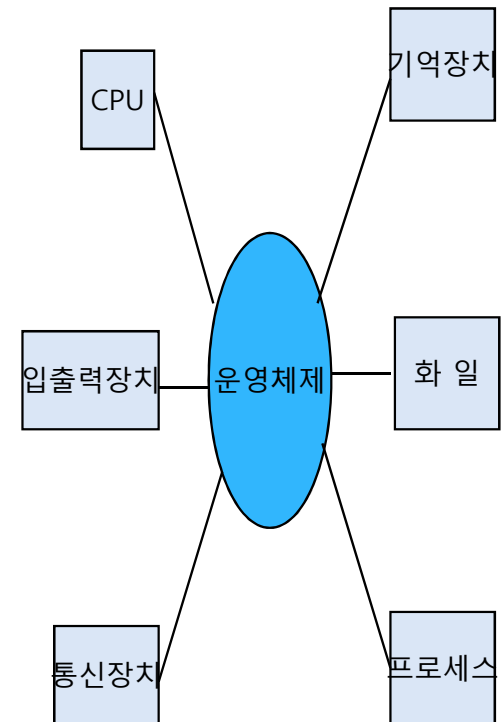
- 입출력 시스템을 관리하기 위해 임시 저장(Buffer-aching) 시스템, 일반적인 장치 드라이버 인터페이스, 특정 하드웨어 장치를 위한 드라이버 기능

■ 파일 관리

- 파일과 디렉터리의 생성과 제거, 보조기억장치에 있는 파일의 맵핑, 안전한(비휘발성) 저장 매체에 파일 저장하는 기능

■ 기타 기능

- 시스템 보호, 네트워킹, 명령어 해석기와 시스템 관리 기능



운영체제의 구성

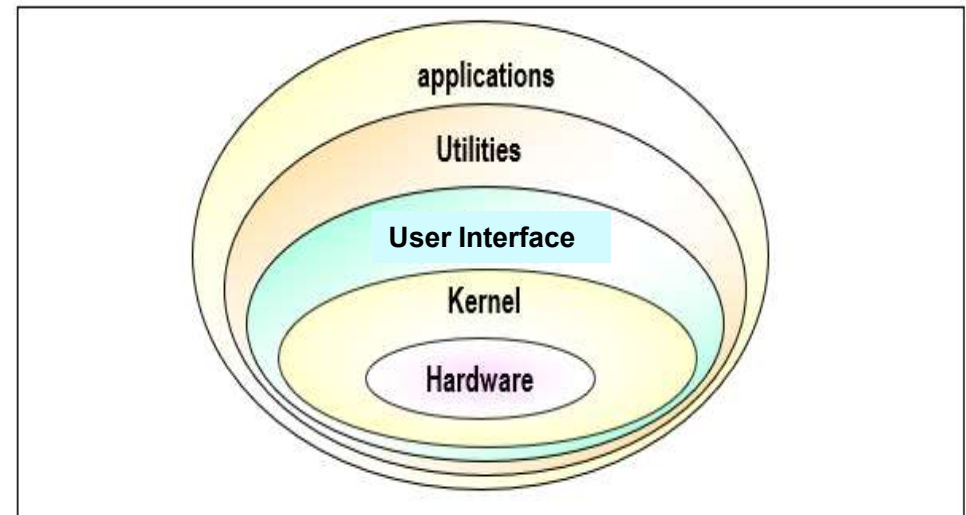
• 운영체제의 구성

■ 커널 (kernel)

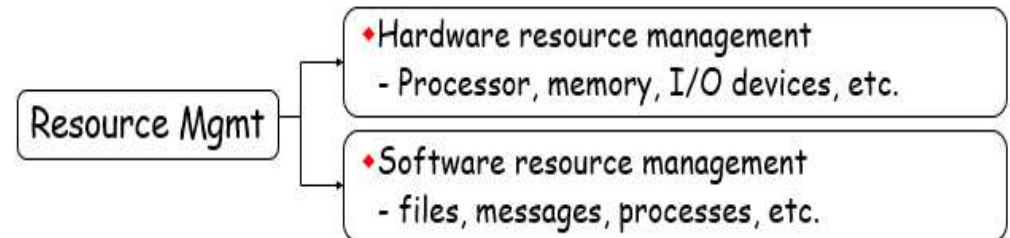
- OS의 핵심부분 (booting 후 memory에 상주)
- 사용자 및 실행 프로그램들을 위해 빈번히 사용되는 기능 담당
- 주로 자원의 관리 기능
- 동의어
 - 핵 (neucleus)
 - 관리자 (supervisor) 프로그램,
 - 상주 프로그램 (resident program)
 - 제어 프로그램 (control program)

■ 유틸리티 프로그램 (utility programs)

- 비상주 프로그램, 서비스 프로그램
- 주로 사용자 인터페이스 기능



커널의 주요 기능



운영체제의 자원 관리

- 운영체제의 자원 관리

- 프로세스(process) 관리
=> 프로세스의 생성과 삭제, 중지와 계속 등을 관리
- 작업(job) 관리
=> 작업과 관련된 순서, 우선순위, 프로세스 할당 등 관리
- 주기억 장치 관리
=> 주기억 장치의 할당과 회수를 담당함
- 보조기억 장치 관리
=> 보조기억 장치의 효율적인 사용을 관리
- 입출력 장치 관리
=> 입출력 장치들을 관리함
- 파일 관리
=> 파일의 관리를 담당함
- 보안 관리
=> 보안과 관련 된 사항들을 관리함

