

스마트시스템 운영체제 (LD01600)

김준철

정보시스템공학과

greensday@sungshin.ac.kr

1주차 강의

	주차	강의 목차
	9.2	1 과목소개 / 운영체제 개요
	9.9	2 컴퓨터 시스템 구조
	9.16	3 프로세스와 스레드1
휴강(9.30) (추석)	9.22	4 프로세스와 스레드2, CPU스케줄링1
	10.7	5 CPU스케줄링2
	10.14	6 프로세스 동기화
	10.21	7 교착 상태
	10.28	8 중간고사
	11.4	9 물리 메모리 관리
	11.11	10 가상메모리 기초
	11.18	11 가상메모리 관리
	11.25	12 입출력시스템1
	12.2	13 입출력시스템2, 파일시스템1
	12.9	14 파일시스템2
	12.16	15 기말고사

Operating Systems

Ch.01 운영체제의 개요

01 운영체제 소개

02 운영체제의 역사

왜 운영체제를 공부하는가?



차이점?

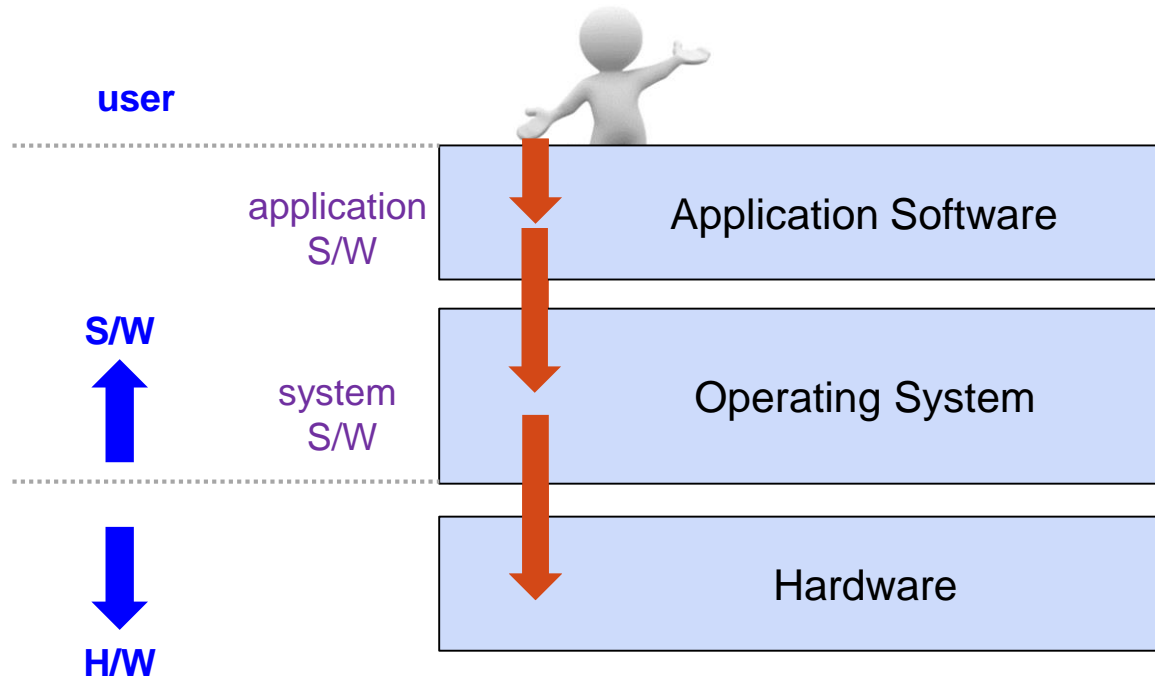


현대의 복잡한 시스템들을 누가 관리해 줄 것인가?

운영체제의 개념

■ 운영체제(Operating System, OS)의 개념

- 컴퓨터 하드웨어 바로 위에 설치되어 **소프트웨어와 하드웨어, 사용자를 연결해 주는 소프트웨어 계층**
 - **사용자** : 컴퓨터를 사용하는 사람이나 장치, 다른 컴퓨터 등을 의미
 - **소프트웨어** : 컴퓨터의 기능 수행에 필요한 모든 프로그램
 - **하드웨어** : 기본 연산자원을 제공하는 프로세서(혹은 CPU), 메모리, 주변장치 등
processor Central Processing Unit



운영체제의 역할

■ 자원 관리

performance 향상

- 컴퓨터 시스템의 자원을 응용 프로그램에 효과적으로 분배하여 사용자가 원활하게 작업할 수 있도록 함
- 자원을 요청한 프로그램이 여러 개라면 적당한 순서로 자원을 배분하고 적절한 시점에 자원을 회수하여 다른 응용 프로그램에 나누어줌

■ 자원 보호

- 비정상적인 작업으로부터 컴퓨터 자원을 보호

■ 하드웨어 인터페이스 제공

convenience 향상

- 사용자가 복잡한 과정 없이 다양한 장치를 사용할 수 있도록 해주는 하드웨어 인터페이스 제공
- CPU, 메모리, 키보드, 마우스와 같은 다양한 하드웨어를 일관된 방법으로 사용할 수 있도록 지원

■ 사용자 인터페이스 제공

- 사용자가 운영체제를 편리하게 사용하도록 지원
(예: 윈도우의 그래픽 사용자 인터페이스(GUI))

운영체제의 역할 - 자원 관리

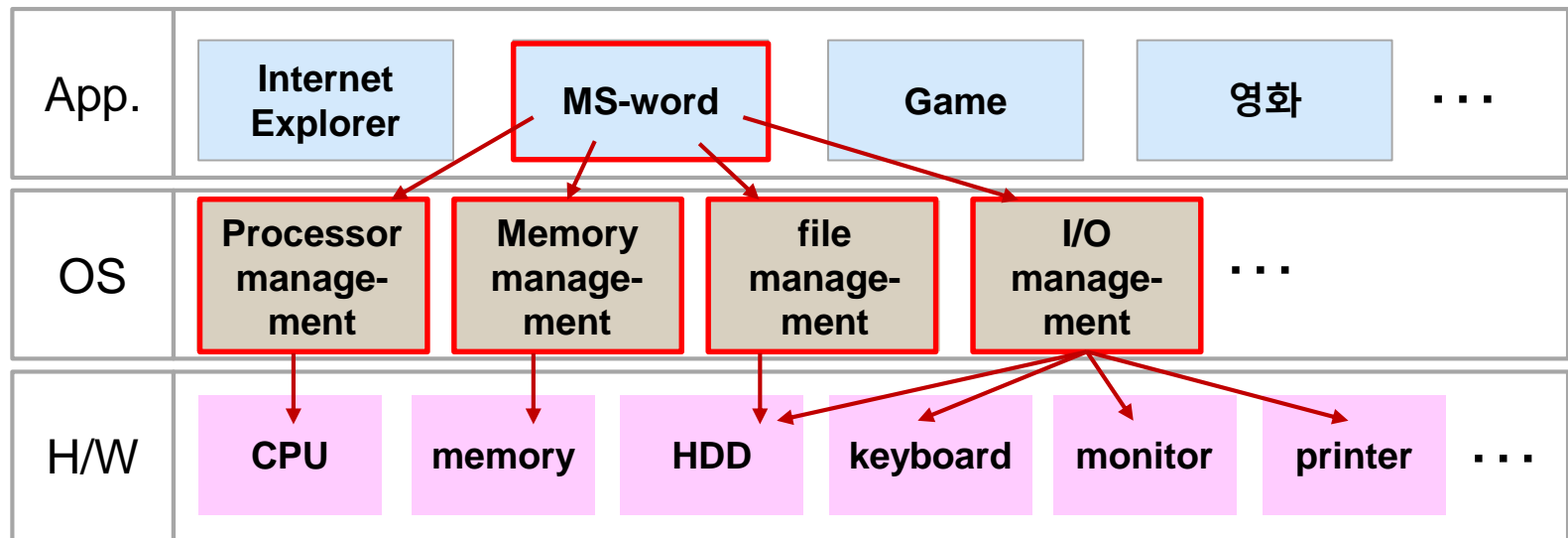
■ 자원 관리

- 자원을 효율적으로 사용할 수 있게 함
 - Resource management (자원 관리)
 - Resource allocation (자원 할당)

자원

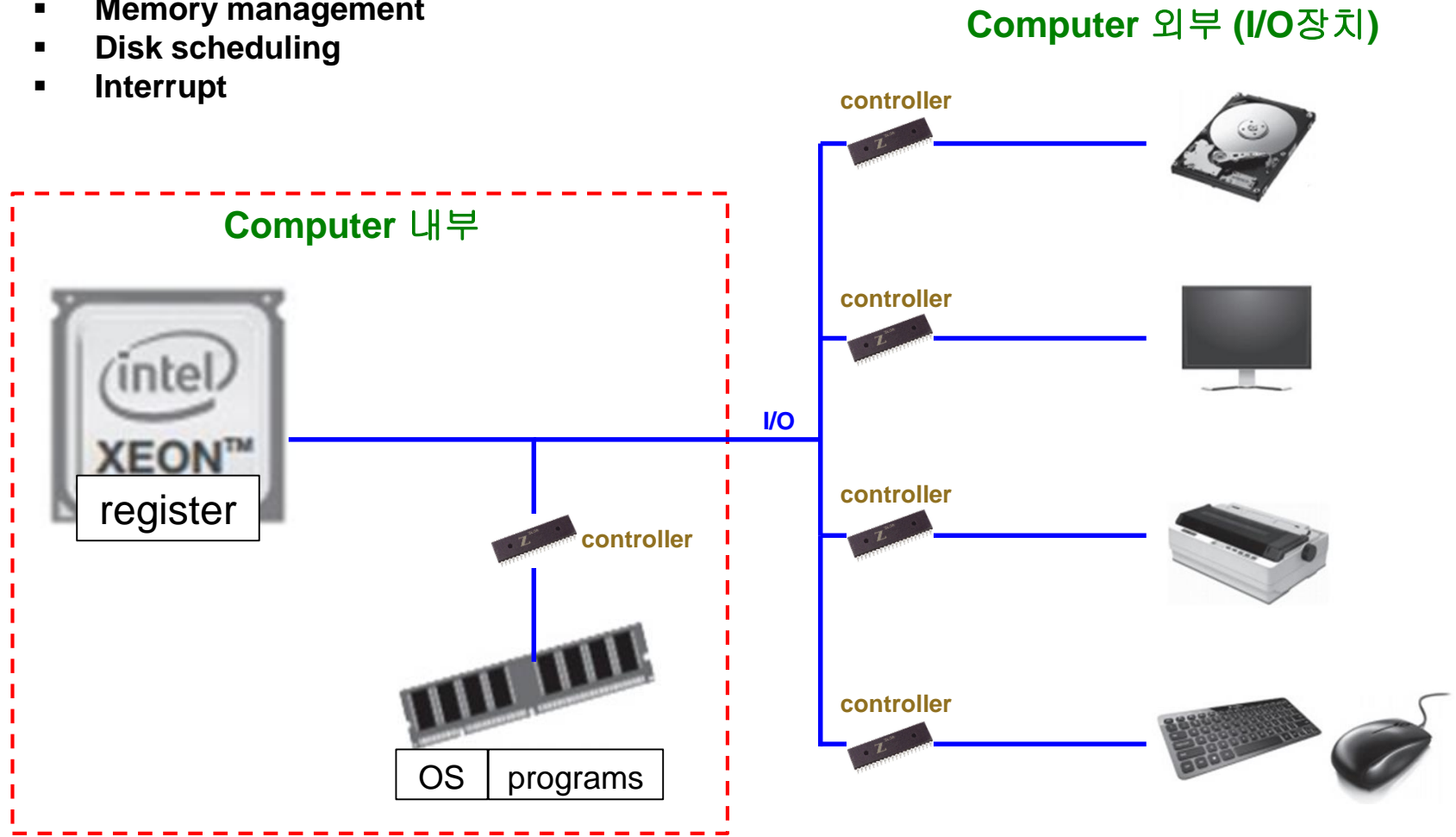
H/W자원 : CPU, Memory, HDD,...

S/W자원 : process, file,...



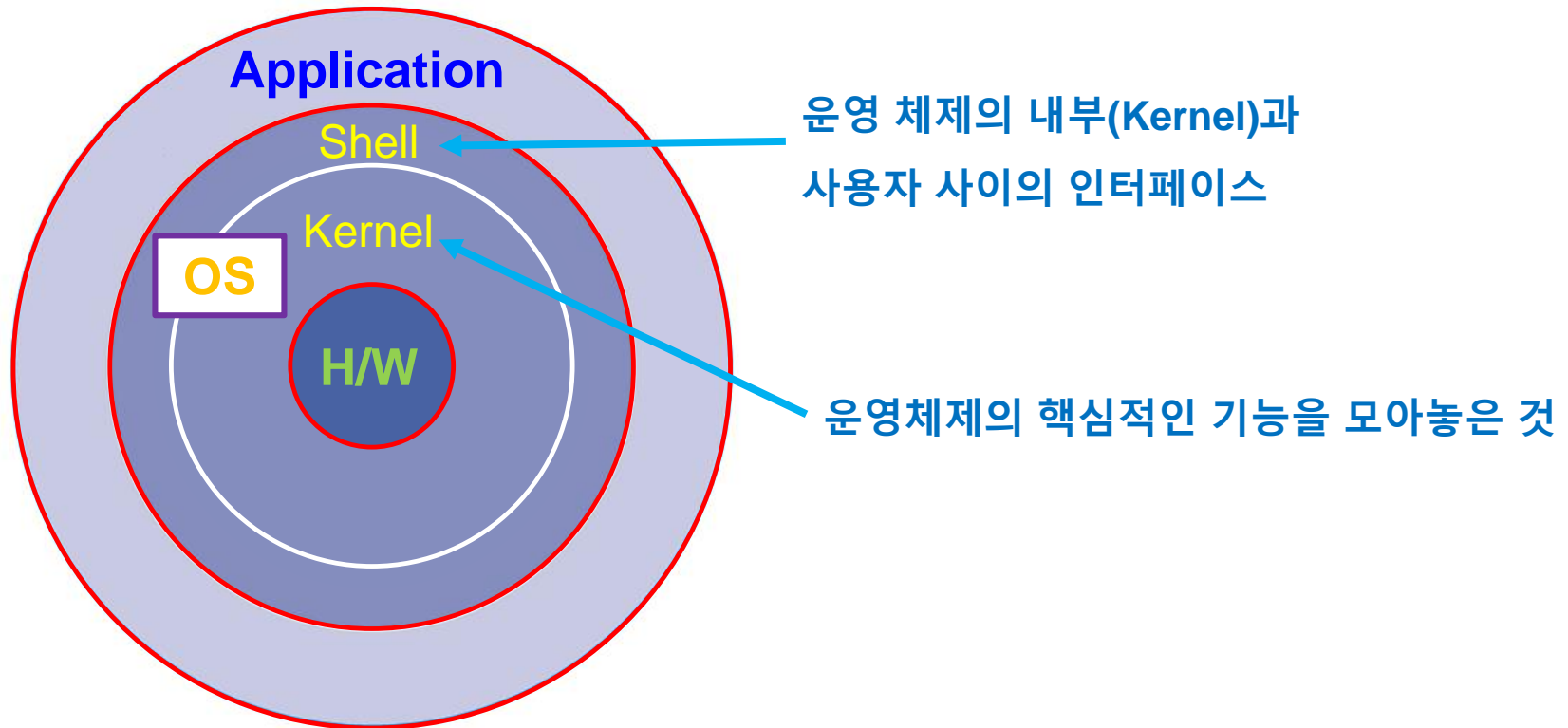
컴퓨터 구조 및 운영체제의 역할

- CPU scheduling
(Processor management)
- Memory management
- Disk scheduling
- Interrupt



커널과 셸

- 커널(Kernel)
 - OS중 자원을 관리하는 부분
- 셸(Shell) - command interpreter
 - OS중 User의 명령을 받아서 해석하여 실행하는 부분



운영체제의 목표

- 운영체제의 목표



운영체제의 목표

■ 효율성

- 자원을 효율적으로 관리하는 것
- 같은 자원을 사용하여 더 많은 작업량을 처리하거나, 같은 작업량을 처리하는 데 보다 적은 자원을 사용하는 것

■ 안정성

- 작업을 안정적으로 처리하는 것
- 사용자와 응용 프로그램의 안전 문제와 하드웨어적인 보안 문제 처리
- 시스템에 문제가 발생했을 때 이전으로 복구하는 결함 포용 기능 수행

■ 확장성

- 다양한 시스템 자원을 컴퓨터에 추가하거나 제거하기 편리한 것

■ 편리성

- 사용자가 편리하게 작업할 수 있는 환경을 제공하는 것

운영체제의 역사

■ 주변 기기의 발전을 중심으로 살펴본 운영체제의 역사

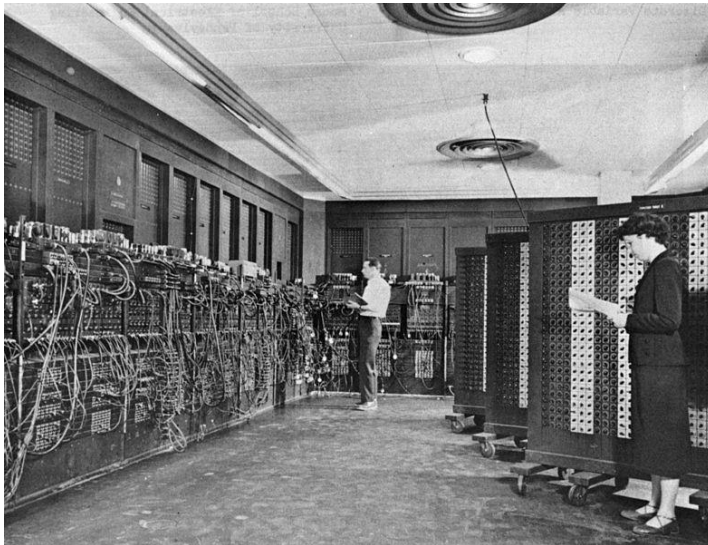
[표] 운영체제의 역사

구분	시기	주요 기술	특징
0기	1940년대	없음	• 진공관(0과 1) 사용
1기	1950년대	카드 리더, 라인 프린터	• 일괄 작업 시스템 • 운영체제의 등장
2기	1960년대 초반	키보드, 모니터	• 대화형 시스템
3기	1960년대 후반	C 언어	• 다중 프로그래밍 기술 개발 • 시분할 시스템
4기	1970년대	PC	• 개인용 컴퓨터의 등장 • 분산 시스템
5기	1990년대	웹	• 클라이언트/서버 시스템
6기	2000년대	스마트폰	• P2P 시스템(메신저, 파일 공유) • 그리드 컴퓨팅 • 클라우드 컴퓨팅 • 사물 인터넷

초창기 컴퓨터 (1940년대)

■ 에니악 (ENIAC)

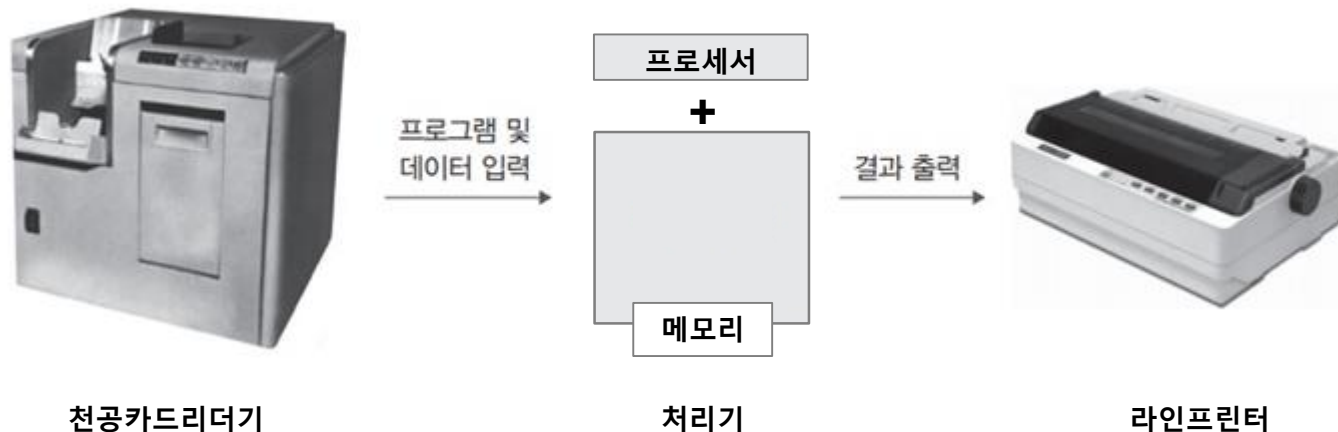
- 백열전구 같은 모양의 **진공관**이라는 소자를 사용하여 진공관이 켜지면 **1**, 꺼지면 **0**이라고 판단
- 전선을 연결하여 논리회로를 구성하는 ‘**하드 와이어링**’ 방식으로 동작
→ 전선 연결 변경으로 프로그래밍 – **외부 프로그램 방식**
(cf. 폰 노이만 구조 – 내부프로그램 방식)
- 운영체제가 없음



[그림] 에니악

발전된 컴퓨터 (1940~1950년대 초반)

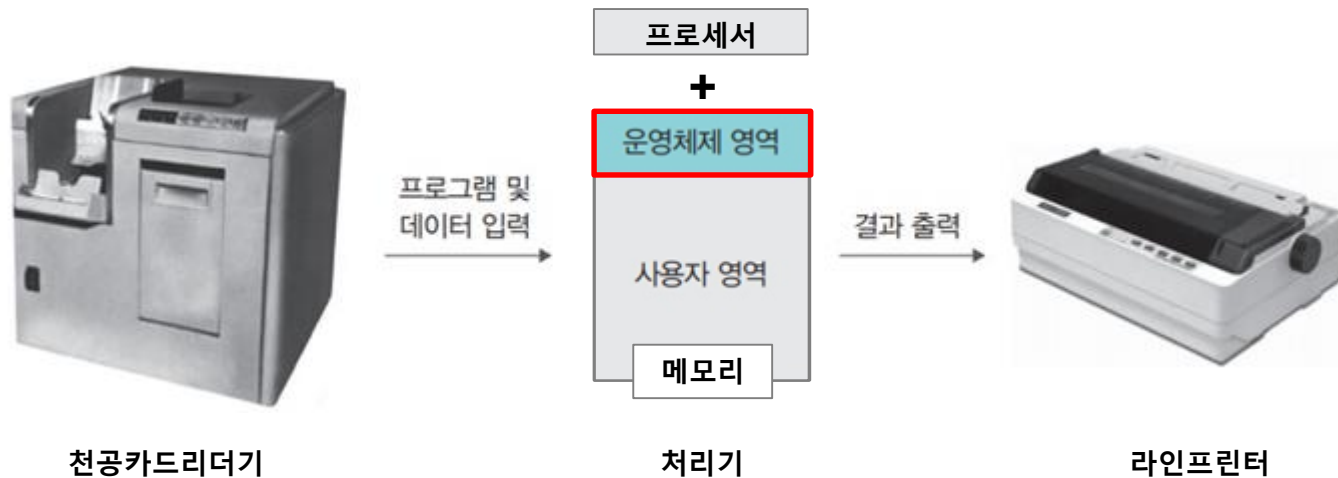
- 운영체제가 없는 시스템 (no operating system)
 - 천공카드리더기(입력)와 라인프린터(출력) 사용 :
모든 작업을 한꺼번에 처리하고 프로그램 실행 중간에 사용자가 데이터를 입력하거나 수정하는 것이 불가능한 시스템
 - 규모가 매우 큼(건물 안에 시스템을 구축)
 - 운영자(operator)가 컴퓨터 시스템을 관리함 (운영체제 프로그램 없음)



[그림] 운영체제가 없는 컴퓨터 시스템

일괄 작업(처리) 시스템 (1950년대)

- 일괄 작업(처리) 시스템 (batch processing system)
 - resident monitor 사용 :
 - 운영체제 – resident monitor
 - 메인메모리가 운영체제의 상주 영역과 사용자의 영역으로 나뉨



[그림] 일괄 작업 시스템의 구성

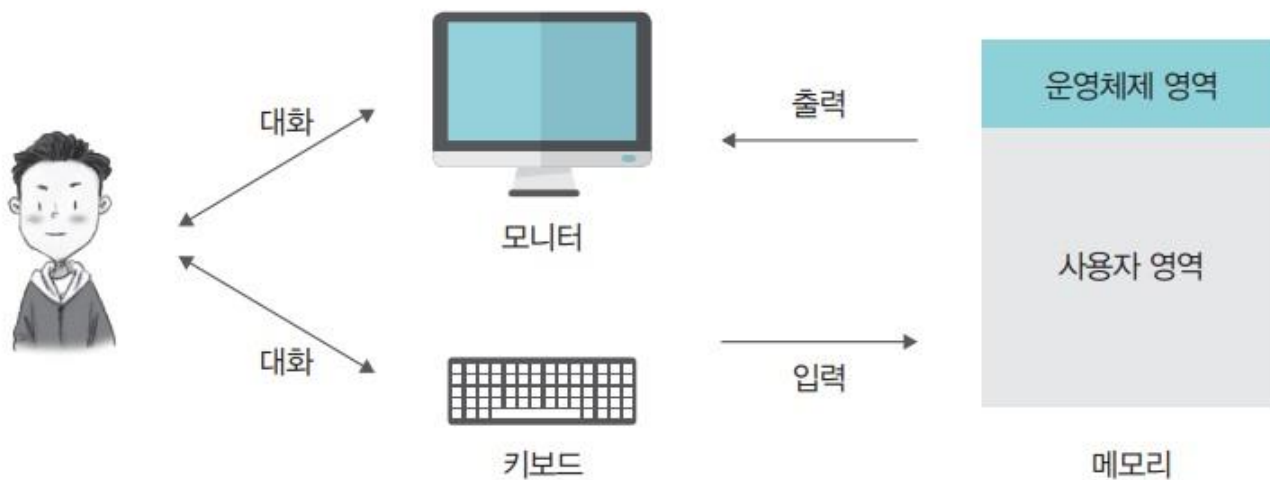
대화형 시스템 (1960년대 초반)

■ 대화형 시스템 (interactive system)

■ 모니터와 키보드의 등장 :

프로그램이 진행되는 도중에 사용자로부터 입력을 받을 수 있어 입력값에 따라 작업의 흐름을 바꾸는 것이 가능한 시스템

■ 대화형 시스템의 등장으로 문서 편집기, 게임과 같은 다양한 종류의 응용 프로그램을 만들 수 있게 됨



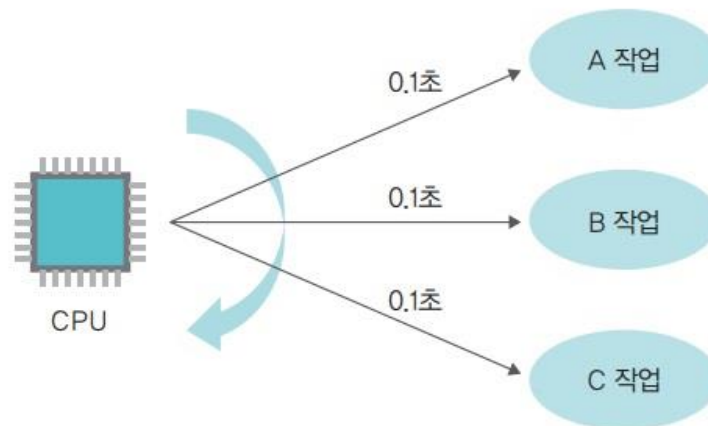
[그림] 대화형 시스템의 구성

다중 프로그래밍 시스템 (1960년대 중후반)

- 다중 프로그래밍 시스템(multiprogramming system)
 - 다중 프로그래밍: 하나의 CPU로 여러 작업을 동시에 실행하는 기술
 - 다중 프로그래밍 시스템: 일괄 작업 시스템에 비해 효율성이 뛰어남
 - 하나의 메모리에 여러 프로그램들이 적재되어 실행되는 시스템
 - 프로그램 수행 중 I/O를 해야 CPU 사용권한이 다음 프로그램으로 넘어감
 - OS에서 고려해야 할 사항들이 많아짐
(CPU scheduling, Memory management, ...)

시분할 시스템 (1960년대 후반)

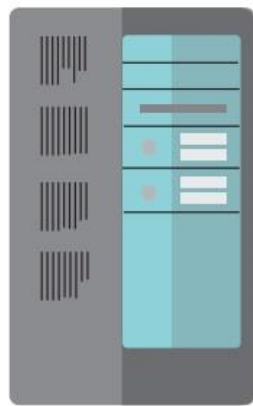
- 시분할 시스템(time sharing system)[=다중작업 시스템(multitasking system)]
 - Interactive computer(상호 대화형 컴퓨터)를 사용하기 시작
→ computer에 monitor, keyboard가 연결되기 시작함
 - 프로그램에 CPU 사용시간을 정해주고, 정해진 시간이 넘으면 프로그램이 다 끝나지 않아도 CPU 사용 권한을 뺏아서 다음 프로그램에게 넘겨줌
 - 정해진 CPU 사용시간의 조각을 ‘타임 슬라이스’ 또는 ‘타임 쿼텀’이라고 함
 - 타임 슬라이스가 매우 짧으면 작업이 동시에 처리되는 것처럼 보임
 - 오늘날의 컴퓨터에는 대부분 시분할 시스템이 사용됨



[그림] 다중 프로그래밍 시스템의 시간 분배

분산 시스템 (1970년대 후반)

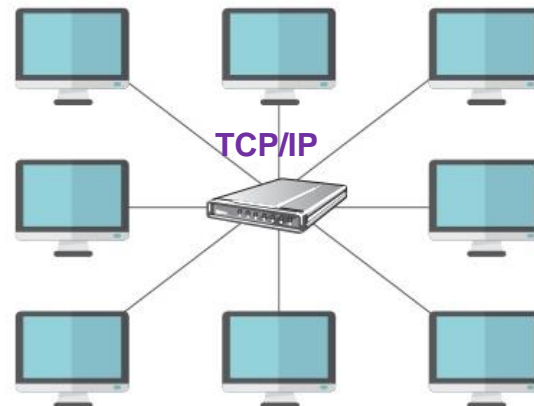
- 분산 시스템 (distributed system)
 - 개인용 컴퓨터와 인터넷이 보급되면서 값이 싸고 크기가 작은 컴퓨터를 하나로 묶어서 대형 컴퓨터(메인프레임)의 능력에 버금가는 시스템을 만들 수 있게 됨
 - 네트워크상에 분산되어 있는 여러 컴퓨터로 작업을 처리하고 그 결과를 상호 교환하도록 구성한 시스템



(a) 메인프레임



성능이 비슷



(b) 분산 시스템

[그림] 메인프레임과 분산 시스템

클라이언트/서버 시스템 (1990년대~현재)

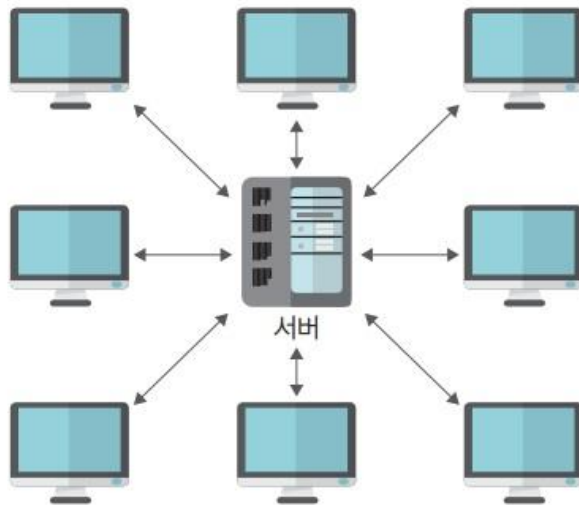
- 클라이언트/서버 시스템 (client / server system)
 - 작업을 요청하는 클라이언트와 거기에 응답하여 요청 받은 작업을 처리하는 서버의 이중구조로 나뉨
 - 웹 시스템이 보급된 이후 일반인들에게 알려짐



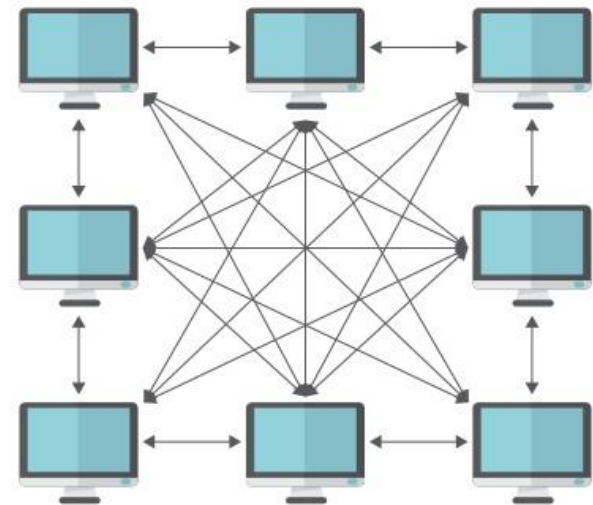
[그림] 클라이언트 / 서버 구조

P2P 시스템 (2000년대 초반~현재)

- P2P 시스템 (Peer-to-peer system)
 - 클라이언트/서버 구조의 단점인 서버 과부하를 해결하기 위해 만든 시스템
 - 서버를 거치지 않고 사용자와 사용자를 직접 연결
 - 냅스터(mp3공유 시스템)에서 시작하여 현재 메신저나 토렌트 시스템 에서 사용



(a) 클라이언트 / 서버 시스템



(b) P2P 시스템

[그림] 클라이언트 / 서버시스템과 P2P 시스템



감사합니다.

