

CHAPTER

# 04

## 운영체제와 시스템 소프트웨어

4.1 운영체제의 개념과 발전

4.2 운영체제의 구성

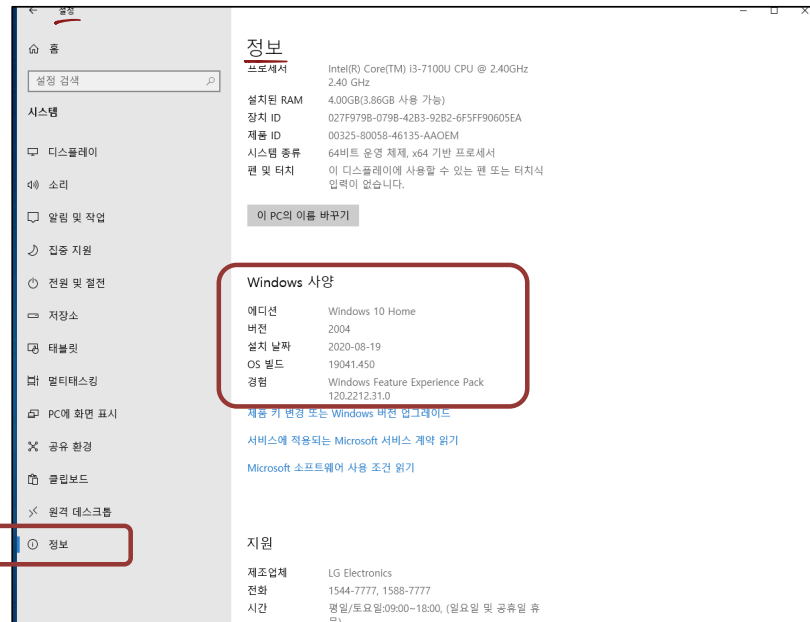
4.3 컴퓨터 실행의 제어

4.4 시스템 소프트웨어

# 4.1 운영체제의 개념과 발전



- 운영체제의 목적과 발전
- 플랫폼 개념

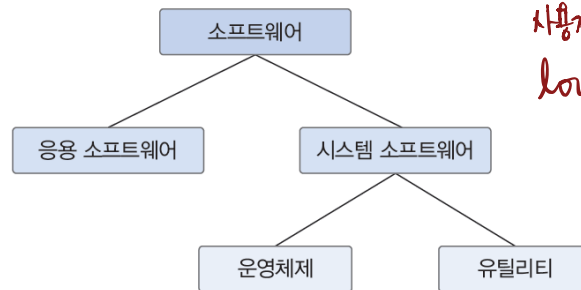


### 4.1.1 운영체제의 목적과 발전



#### ■ 컴퓨터 소프트웨어 분류

- 응용 소프트웨어 (system software 위에서)
  - 개인이나 기관이 개발한 프로그램
  - 예) 포토샵, 일러스트레이터, 엑셀, 파워포인트 등의 상업적 소프트웨어
- 시스템 소프트웨어
  - 운영체제, 파일 압축 소프트웨어, 보안 소프트웨어, 컴파일러 등



사용자들이 하드웨어를 제어할 수 있게 함.  
low level control

| 그림 4-1 소프트웨어의 분류

### ■ 운영체제(Operating System, OS) 정의


- 컴퓨터 시스템을 올바르게 작동시키고 효율적으로 활용하기 위한 가장 핵심적인 소프트웨어
- 하드웨어와 사용자(프로그램이나 이용자)간의 인터페이스 역할 응용
- 모든 이용자와 응용프로그램이 의존하고 있는 기반이 되는 핵심 소프트웨어



### ■ 운영체제의 목적

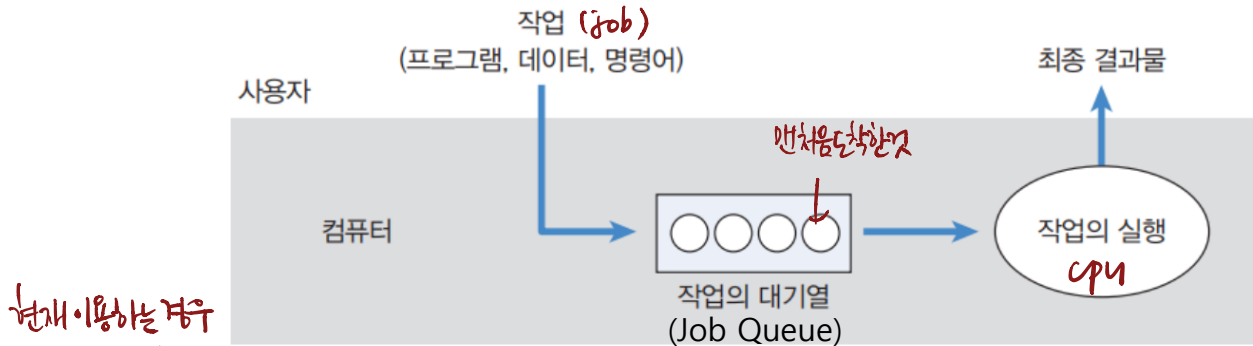
- 1) 하드웨어 자원의 효율적 이용 *OS가 없으면 하드웨어를 직접 관리해야 함*
  - 컴퓨터 시스템의 프로세서, 주기억장치, 보조 기억장치, 입출력장치, 컴퓨터 네트워크와 같은 자원의 효율적 관리
  - 다수 사용자에게 하드웨어 자원의 효율적 배분
- 2) 자원의 편리한 이용
  - 컴퓨터 시스템의 복잡한 자원을 편리하게 사용할 수 있도록
  - 예) Windows 운영체제: 과거 명령어 방식인 MS-DOS에서 GUI 방식으로 지원  
*화면에 나타나는 아이콘 클릭*

### ■ 운영체제의 발전

- 초창기 컴퓨터는 컴퓨터의 모든 작동을 이용자가 수동으로 수행  
*회로, 케이블, ...*
- 그 이후 **오퍼레이터**가 컴퓨터의 작동을 수행 ⇒  
1950년대 **OS**로 발전  

- OS의 처리방식
  - 1) 배치 프로세싱
  - 2) 상호대화식 프로세싱
  - 3) 시분할 시스템
  - 4) 병렬처리
  - 5) 실시간 처리

## 4.1 운영체제의 개념과 발전

- 1) 배치 프로세싱(Batch processing, 일괄처리) <sup>지문도 사용함 (-bat)</sup>
- 컴퓨터 프로그램 흐름에 따라 순차적으로 자료를 처리하는 방식
  - 원칙적으로 도착 순서에 따라 하나씩 실행
    - 처리할 프로그램이나 명령어를 작업 또는 잡(Job)
    - 각 작업은 컴퓨터 시스템 내에서 작업 대기열을 구성하여 자신이 처리될 순서를 기다림
- FIFO(First-In First-Out) 방식

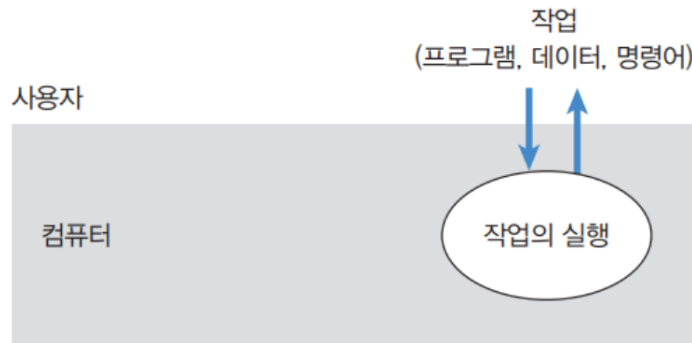


| 그림 4-2 배치 프로세싱의 개념

현재 이용가능 여부  
업무시스템 - 원가, 요금 계산  
알정할 시간 단위로 계산하는 것이  
이용. 실시간 확인이 필요하지 않은 경우

### 2) 상호대화식 프로세싱(Interactive processing)

- 배치 프로세싱은 프로그램 수행 도중 이용자와 컴퓨터 간 상호작용 할 수 없음
- 모니터를 통해 프로그램의 실행 도중에 데이터를 제공하거나 프로그램을 제어, 경우에 따라서 중간 결과를 확인
- 실시간 처리가 요구됨



| 그림 4-3 상호대화식 프로세싱의 개념

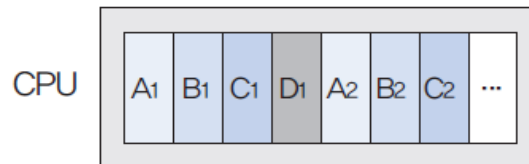


### 3) 시분할 시스템(다중 프로그래밍, 다중작업)

터미널 이용  
PC는 single

- 다중 프로그래밍(multiprogramming): 다수의 사용자가 여러 개의 프로그램을 동시에 실행 메인프레임이나 워크스테이션
- 다중 작업(multitasking): 다수의 작업이 CPU와 같은 공유자원을 나누어 사용
- CPU가 하나인 경우, CPU 시간을 분할하여 순서대로 돌아가며 실행 → 시분할(Time Sharing) 기법
  - 하나의 CPU가 특정 순간에는 하나의 태스크만 수행할 수 있으나, 사용자에게는 병렬연산이 이루어지는 것 같은 환경 제공

예) 4개의 프로그램 A, B, C, D

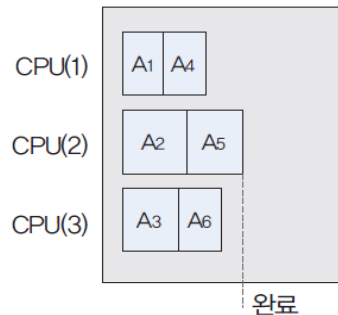


| 그림 4-4 다중작업의 개념

### 4) 병렬처리(Parallel System) 최소

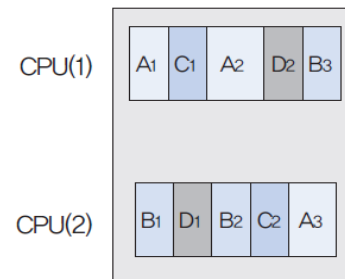
- 하나의 프로그램 A를 여러 개의 작업  $\{A_1, A_2, \dots, A_n\}$ 으로 분할하여 몇 개의 CPU에 할당하여 빠르게 처리하는 방식 : 하드웨어적으로 구현
- 다중 처리(Multiprocessing): CPU 효율을 극대화 하기 위한 방법으로 여러 사용자 프로그램이 동시에 수행되는 것처럼 처리하는 방법 : 소프트웨어적으로 구현
  - 다수의 프로세서를 활용하거나 시분할처리를 수행

병렬처리



(a) 병렬처리

다중처리



(b) 다중처리

병렬처리에 관련된지는 않음  
여러개의 CPU를 사용한다고 생각

사용자가 원하는 짧은 시간

### 5) 실시간 처리(Real-Time Processing)

과거에는 프세서 (하드웨어) 자체가 처리시간이 길지 않기 때문에 중요하지 않음

- 처리를 요구하는 자료가 발생할 때마다 **즉시** 처리하여 정해진 짧은 시간 내에 작업을 완료하여 응답하는 방식
- 보통 주어진 시간내에 작업을 마칠 수 있으면 실시간 시스템 IoT

'짧은 시간'의 개념이 강화됨.

예)

- 자율자동차의 경우 센서들을 통해서 들어오는 외부 세계의 데이터
- 동영상의 비디오 부분과 오디오 부분의 실시간 처리

### ■ 운영체제의 역사 (1)

- 초창기의 운영체제 1950년대

- 컴퓨터 시스템마다 자체 운영체제
- 예) IBM OS/360, CDC SCOPE

- 유닉스(UNIX) 운영체제 *현재의 운영체제 개념 확립*

- 1969년 AT&T사 벨 연구소의 토마스 리치 등이 어셈블리어로 개발, 1971년 C 언어로 다시 작성
- 다중사용자 및 다중작업 기능 지원
- 명령어 방식의 인터페이스 (MS-DOS 같은 형태)
- 유닉스 계열 운영체제 : HP-UX, 솔라리스(Sun Microsystems)

*동작이함*

### ■ 운영체제의 역사 (2)

#### • 리눅스(Linux)

- UNIX 계열의 운영체제 중 대표적인 공개 소프트웨어, 1991년 리누스 토발즈(Torvalds)가 개발
- 최초의 공개소스 운영체제의 커널로 개발
  - 공개 소프트웨어 운동 : MIT 리차드 스톨만(Stallman)교수가 GNU를 개발하면서 시작, '오픈 소스(Open Source)' 용어 탄생
  - GNU 운영체제는 커널이 개발되지 못함.
- 리눅스 배포 : 레드햇, 우분투, 데비안, 페도라 등

↓  
커널이  
구현됨  
외로



(a) GNU 로고



(b) Linux 로고



(c) RedHat 로고

| 그림 4-6 Linux 관련 로고

### ■ 운영체제의 역사 (3)

- 데스크탑 운영체제 *PC의 등장 이후.*

- DOS, MS Windows, Mac OS, UNIX, Linux 등
- GUI 방식: 1984년 Mac OS, 1980년대 후반 MS Windows *해킹토시*  
*나 첫때에는 명령어 방식이었음*

```
Welcome to FreeDOS

CuteMouse v1.9.1 alpha 1 [FreeDOS]
Installed at PS/2 port
C:\>over

FreeCom version 8.82 pl 3 XMS_Swap [Dec 18 2003 06:49:21]

C:\>dir
Volume in drive C is FREEDOS_C95
Volume Serial Number is 0E4F-19EB
Directory of C:\

FDOS      <DIR>    08-26-04   6:23p
AUTOEXEC  BAT       435  08-26-04   6:24p
BOOTSECT  BIN       512  08-26-04   6:23p
COMMAND   COM     93,963  08-26-04   6:24p
CONFIG    SYS       801  08-26-04   6:24p
FDOSBOOT  BIN       512  08-26-04   6:24p
KERNEL    SYS     45,815  04-17-04   9:19p
        6 file(s)    142,838 bytes
        1 dir(s)    1,064,517,632 bytes free

C:\>
```

DOS의 명령줄 인터페이스



윈도우 실행화면  
(GUI형식)

### ■ 모바일 운영체제

- 컴퓨터에 비하여 장치 규모가 작으며 각 기기의 특수한 상황에서의 기능을 주로 요구하므로 운영체제의 핵심 기능만 필요 + 전력소모와 저장공간을 줄이는 것이 중요
- 임베디드 장치에 필요한 시스템 소프트웨어를 임베디드 운영체제라고 함
  - 고사양의 임베디드 운영체제로 Windows CE와 임베디드 Linux
  - PDA나 휴대폰 외에 셋톱박스, 로봇, 내비게이션 시스템, 티켓머신 등에서 사용
- 초창기 스마트폰: Symbian OS, Windows mobile
- 2000년 이후: 애플의 iOS, 구글의 개방형 모바일 운영체제 안드로이드(Android)

### 4.1.2 플랫폼 개념



#### ■ 플랫폼이란? *platform*

- 어떤 일을 하기 위해 필수적으로 거쳐야 하는 장소를 의미
- 운영체제는 필수적인 핵심 소프트웨어로 일종의 플랫폼 역할
- "기반" 또는 "매개"의 개념

*여러가지로 정의할 수 있음.*

*하드웨어, 소프트웨어 모두 포함*



### ■ ICT 분야의 플랫폼

- 예) MS사의 윈도우, 구글, 애플, 아마존, 페이스북, 이베이, 네이버, 카카오, 라쿠텐, 알리바바 등
- 플랫폼이란 어떤 목적을 위한 환경을 구축하여 이용자, 개발자, 사업자들이 모여 하나의 생태계를 이룸 *ex. 애플* *하나의, 또는 여러개의*

*\** ( - 개발자나 사업자들이 용이하게 개발하고 사업할 수 있도록 여러 가지 장치와 도구 제공

- “네트워크 효과(Network Effect)” *사용자에게 더 많은 서비스를 제공, 더 많이 모임*
- 플랫폼의 개념은 물리적 세계, 하드웨어, 소프트웨어 및 인터넷 서비스에도 폭넓게 적용

*물리적인 세계에서도 적용됨*

## 플랫폼의 역할

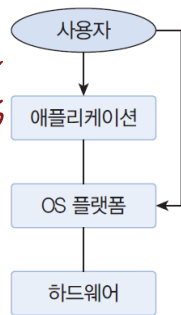
### 1) 기반형 플랫폼

사용자들이 어떤 일하기 위해  
OS를 제공하는 플랫폼

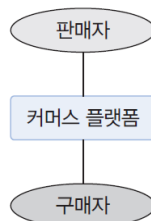
- OS 플랫폼과 같이 하드웨어와 애플리케이션 사이에 존재, API(application Program Interface) 및 SDK(Software Development Kit) 제공 인터페이스(도구) 제공
- 기반형 플랫폼이 애플리케이션 생태계로 자리 잡고 나면 새로운 플랫폼이 끼어들기 어려움

다른 플랫폼에 이용하기 어려움  
ex. 구글 앱을 iOS에 사용X

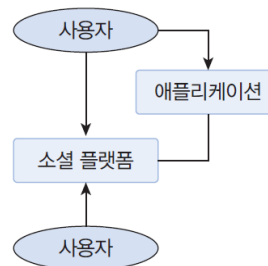
명령력이 큰 OS를  
중심으로 생태계가 이루어짐  
대표적으로 안드로이드/iOS



(a) OS 플랫폼



(b) 커머스 플랫폼



(c) 소셜 플랫폼

| 그림 4-11 플랫폼의 유형과 역할

### 2) 매개형 플랫폼

- 판매자와 구매자가 서로 만날 수 있는 환경 제공, 다수의 구매자와 다수의 판매자가 특정한 목적으로 함께 모이는 장터 *판매자와구매자가만날수있는환경제공*
- 예) 아마존의 오픈 마켓(Open Market), 추천 시스템

### 3) 복합형 플랫폼

*전자상거래시스템*

- 매개형 플랫폼과 기반형 플랫폼의 복합 형태
- 소셜 미디어, SNS 사이트: 매개형으로 출발, API 지원으로 발전 *→기반형의특징도가짐 ex.facebook*
- 모바일 운영체제(안드로이드): 기반형으로 출발, 앱을 통하여 개발자와 사용자가 서로 만나는 장터 제공