

## 4주차 2차시 운영체제의 구성

### 【학습목표】

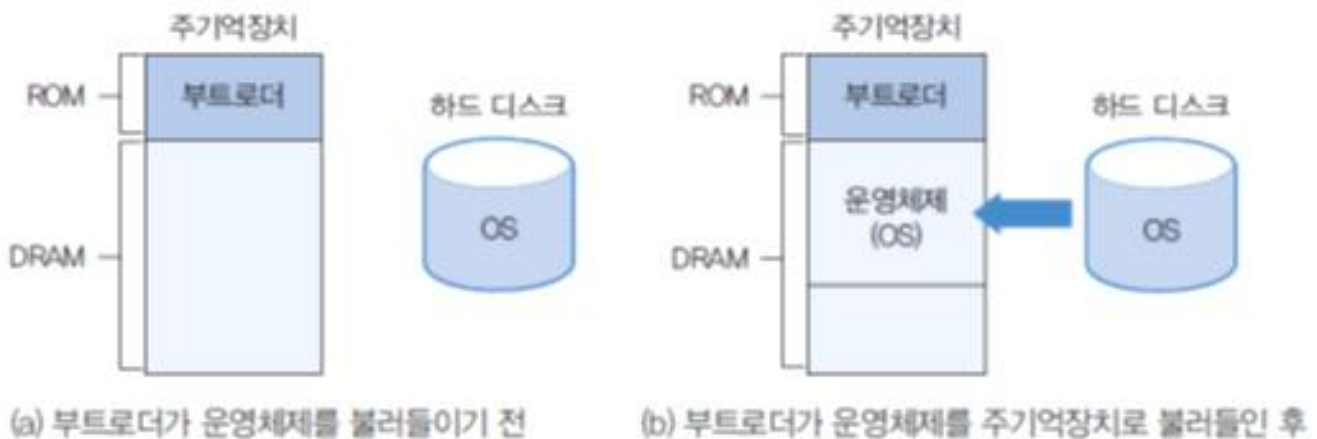
1. 운영체제를 작동하는 방법을 살펴보고 설명할 수 있다.
2. 컴퓨터의 실행과 제어방법을 살펴보고 설명할 수 있다.

### 학습내용1 : 운영체제의 시동

- 자동차를 운행할 때 시동을 거는 것과 같이, 컴퓨터에 전원을 넣으면 운영체제가 시동 됨을 강조

#### 1. 컴퓨터의 시동

- 부트 로더(Boot Loader): 운영체제를 컴퓨터 시스템 내부로 가져와 작동
- 부트 로더는 ROM 메모리에 저장
  - DRAM은 휘발성 기억장치, ROM은 비휘발성
  - ROM에 저장되어 있는 부트 로더는 펌웨어(Firmware)

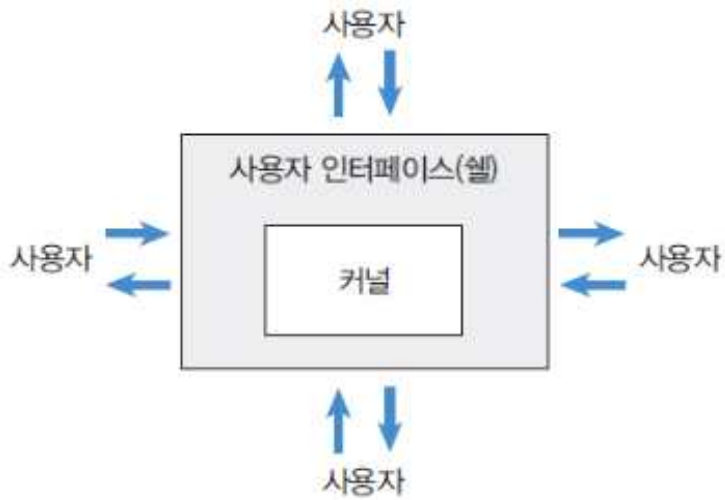


#### 2. 사용자 인터페이스와 커널

- \* 사용자 인터페이스 셸(Shell)
- 컴퓨터 사용자가 컴퓨터 시스템에게 원하는 작업을 요청하기 위한 상호작용을 지원
  - (1) UNIX에서는 사용자 인터페이스를 셸(Shell)
  - (2) GUI 방식의 셸을 윈도우 관리자(Windows Manager)

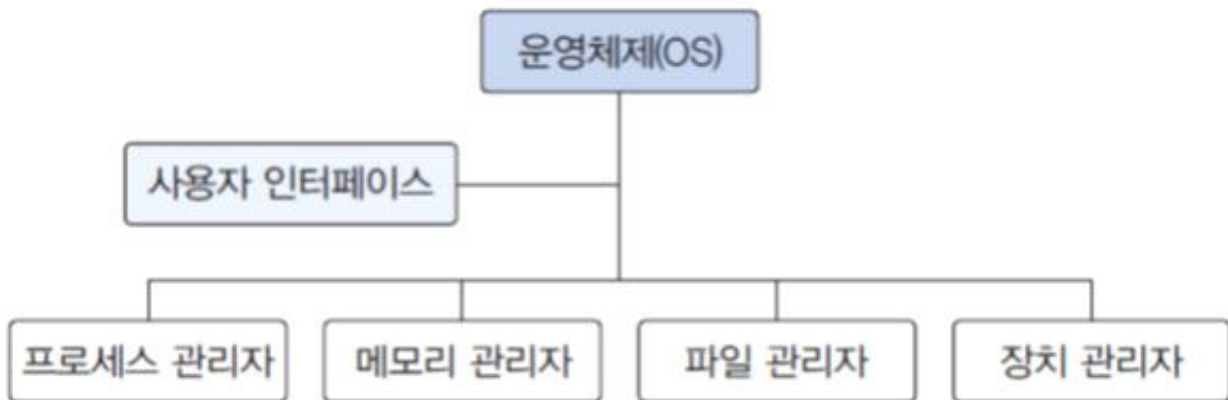
- 사용자 인터페이스의 유형

- (1) 명령어(Command Line) 방식
- (2) 메뉴(Menu Driven) 방식
- (3) 그래픽 사용자 인터페이스(GUI: Graphic User Interface)



\* 커널(Kernel)

- 프로세스 관리자(Process manager) : 프로세스 관리
- 메모리 관리자(Memory manager) : 기억장치 관리
- 파일 관리자(File manager) : 파일 저장, 검색 관리
- 장치 관리자(Device manager) : 컨트롤러 제어 관리



[운영체제(OS)의 구성요소]

### 3. 메모리와 파일 관리

\* 가상 메모리

- 실행 중인 프로그램을 적당한 크기로 나누어 현재 실행에 꼭 필요한 부분을 RAM에 배치하여, RAM의 용량이 훨씬 큰 것처럼 느끼도록 처리

- 가상 메모리 기법

- 페이징(Paging) 기법 : 크기가 일정
- 세그멘테이션(Segmentation) 기법 : 크기가 가변
- 스와핑 : 프로그램 단위로 주기억 장치와 보조기억장치간에 필요시 적재하는 방식

[운영체제의 주기억장치 관리 기법]



- 하드디스크의 파일을 계층적 구조에 따라 디렉토리(Directory)와 폴더(Folder)에 저장

- 파일경로(Directory Path)

- Unix의 예:

C://Introduction to Computers/text/ch03\_software.ppt

- 파일 확장자의 예

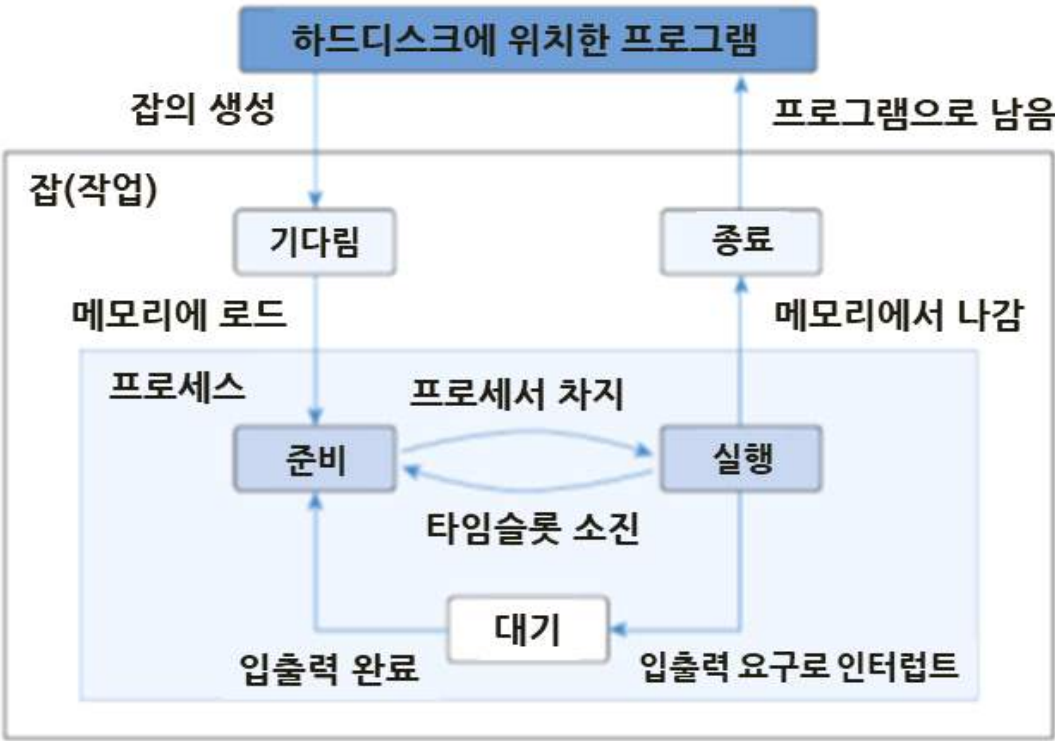
파일의 종류	확장자
프로그램	.exe, .com, .bat
문서	.hwp, .doc, .ppt, .txt, .pdf, .html, .xml
멀티미디어	.bmp, .jpg, .gif, .wav, .au, .mp3, .mpeg
압축	.zip, .alz

학습내용2 : 컴퓨터 실행의 제어

\* 컴퓨터는 모든 일을 몇 개의 처리하는 단위(Process)로 구성해서 일을 처리한 점을 강조

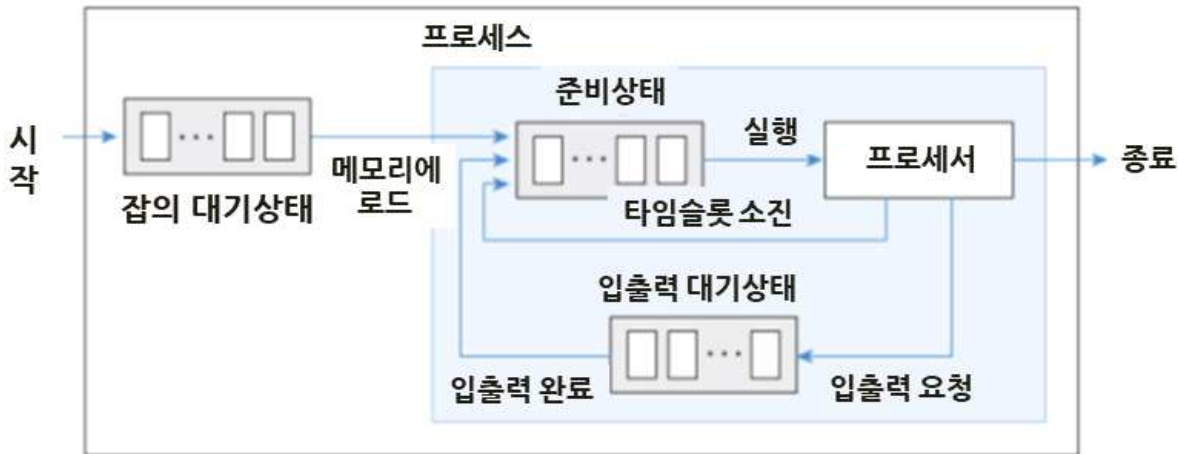
1. 프로세스의 개념

- \* 프로세스(Process)의 개념
  - 프로그램을 실행할 목적으로 생성된 동적 엔티티로 컴퓨터에서 작업을 하는 단위
  - 프로그램, 작업 또는 잡(Job), 프로세서의 차이점
  - 프로세스의 처리단계 : “준비(Ready)”, “실행(Running)”, “인터럽트(Interrupt)”, “대기(Wating)”, “종료(Terminated)” 상태



- 프로세스의 관리
  - 대기 상태(Job Queue)
  - 준비 상태(Ready Queue)
- 프로세스 처리 방식
  - FCFS(First Come First Served) : 적재 순서대로 처리
  - SPN(Shortest Process Next) : 처리시간이 작은 것
  - 시분할(Time Sharing) : 타임슬롯을 “Round-Robin” 방식으로 배정하여 적재

### 잡

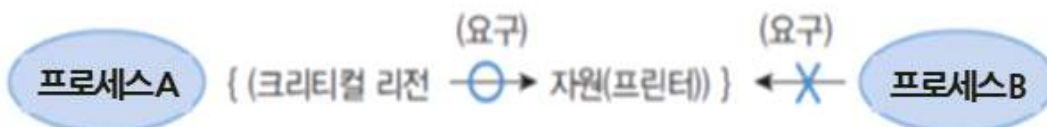


## 2. 컴퓨터 자원의 경쟁

- \* 세마포(Semaphore)
  - 깃발(Flag)의 개념: 깃발을 세마포라 함
    - 예) 프린터의 사용: "Flag = 1"이면 프린터를 이용할 수 있는 상태, "Flag = 0"이면 프린터를 사용할 수 없는 상태
- 크리티컬 리전(Critical Region)
  - 프로그램의 어떤 부분이 시작되어 완료될 때까지 다른 프로세스가 간섭하면 안되는 프로그램의 부분
  - 세마포를 이용하여 크리티컬 리전 표시는 잠금(Lock),작업 완료시 해제(Unlock)

[세마포를 이용한 자원의 할당]

### 세마포 Set(Lock)

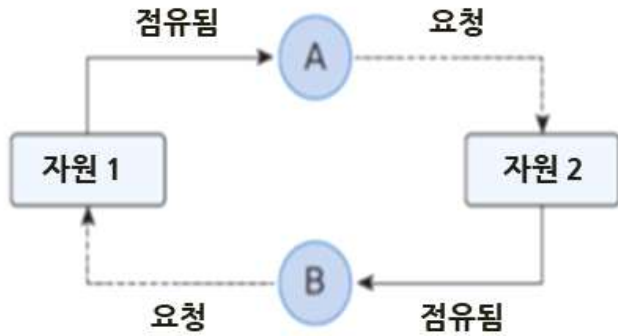


### 세마포 RSet(Unlock)

\* 데드록(Deadlock)

- 프로세스 간에 자원을 차지하려고 경쟁하다 생기는 문제
- 더 이상 프로세스의 처리 작업을 진행할 수 없는 상태가 발생

[데드록의 개념]



(a) 프로세스 간의 데드록

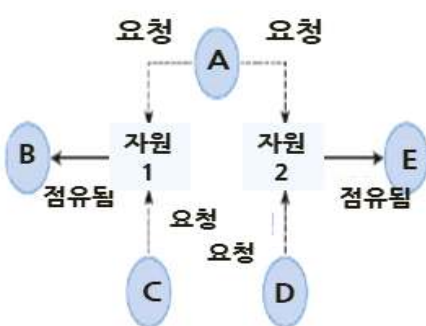


(b) 교통 상황에서 데드록 상태

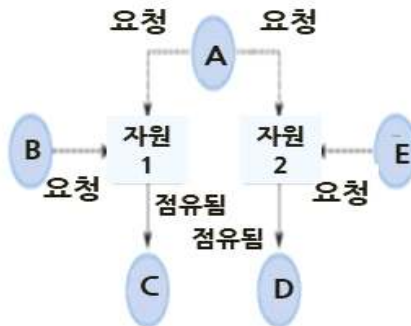
- 데드록 발생 조건
  - 공유할 수 없는 자원의 경쟁이 존재하므로
  - 필요한 자원을 한꺼번에 점유하지 않고 하나씩 점유하기 때문에, 일단 점유한 자원은 필요한 자원을 모두 차지할 때까지 놓아주지 않음
  - 자원이 프로세스에 의해 점유된 후에는 강제로 뺏을 수 없기 때문에
- 데드록 탐지(3가지 발생조건 탐지)와 데드록 회피(발생하지 말도록 통제)
- 스푼링(Spooling) 개념

\* 프로세스 기아(Starvation)

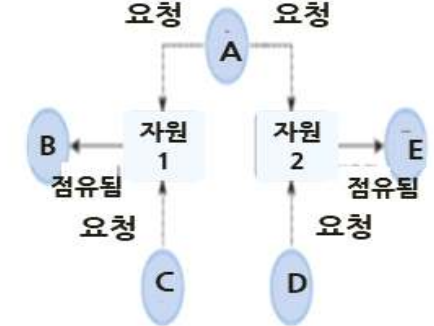
- 데드록이 발생하지 않도록 해도 또 다른 문제가 프로세스의 기아 문제
- 영원히 자원을 차지하지 못하는 프로세스가 발생
- 프로세스 기아 현상의 사례



(a) 프로세스 A가 자원 1 및 자원 2를 요청함



(b) 프로세스 A가 자원 1 및 자원 2를 차지하려고 요청중



(c) 프로세스 A가 아직 자원 1 및 자원 2를 차지하려고 요청중

## 【학습정리】

1. 사용자 인터페이스 유형으로는 명령어 방식, 메뉴방식, 그래픽 사용자 인터페이스 방식이 있다.
2. 주기억장치인 RAM용량이 부족한 경우 하드디스크 일부를 RAM처럼 활용하는 가상메모리 관리 기법을 이용한다.
3. 프로세스 간에 자원을 차지하려는 경쟁 상태를 플래그(Flag)를 이용해서 해결하는 방식을 세마포(Semaphor)라 한다.