인덕대학교

프로세스 관리
- 프로세스의 개요

컴퓨터소프트웨어학과

김병국 교수

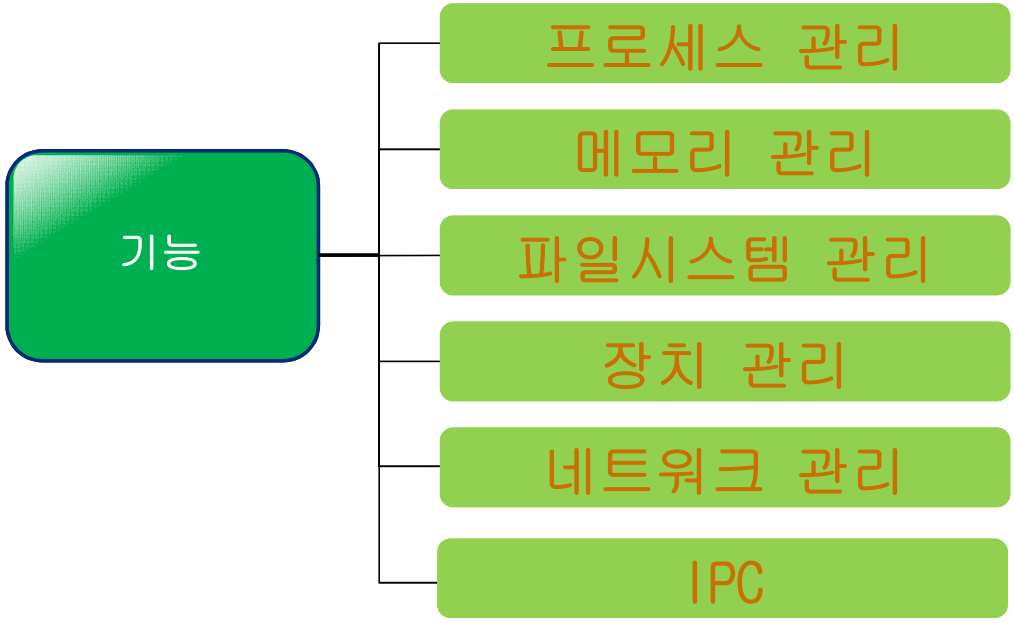
학습목표

- 프로그램과 프로세스의 차이를 안다.
- 프로세스 제어 블록에 대하여 이해한다.
- 프로세스에 대한 상태를 안다.

목차

1. 프로세스의 개념 (1/2)

❑운영체제의 기능

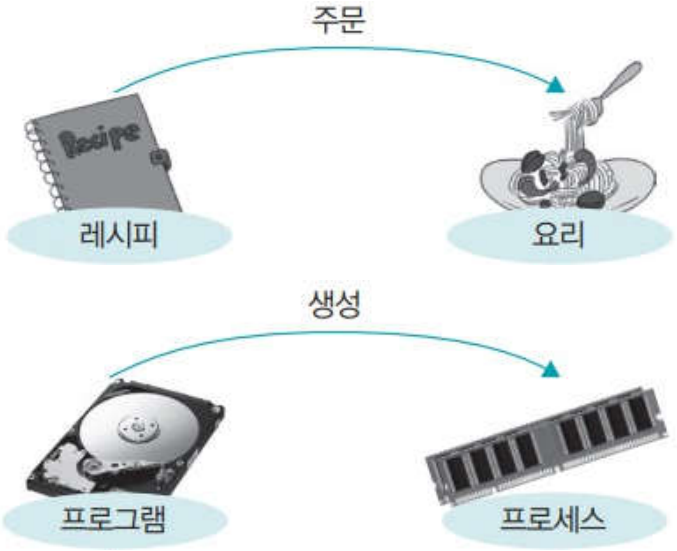


1. 프로세스의 개념 (2/2)

- 프로그램(Program)
  - 저장장치에 기록되어 있는 정적인 소프트웨어
  - 실행 가능한 소프트웨어
    - 컴파일 언어 기반의 소프트웨어일 경우 **main() 함수**가 반드시 존재하는 소프트웨어
    - main() 함수가 없다면, 이는 라이브러리(library) 등으로 분류

- 프로세스(Process)
  - 메모리에 로드(load)되어 주기적으로 CPU에 의해 처리되는 소프트웨어
  - 주기적으로 CPU를 점유
  - **프로세스 제어 블록(PCB: Process Control Block** 또는 Process Status Block 자료구조) 할당
  - 동음어: 태스크(Task), 잡(Job)

- 프로세서(Processor)
  - 명령을 실행(처리)하는 하드웨어
  - 예: CPU, MPU(Micro Process Unit), MCU(Micro Control Unit), AP(App Processor) 등



[요리에 비유한 프로세스]

2. 프로세스 관리

❑ 관리 기능

- 프로세스 등록
- 프로세스 소멸
- 프로세스의 **상태 정의와 전이**(transaction)
- 문맥 교환(Context Switching) 및 스케줄링(Job Scheduling)
- 인터럽트 처리

태스크(Task)

- 시스템에서 취급되는 작업의 단위
- 운영체제에서 프로세스**(Process)** 또는 스레드**(Thread)** 단위

3. 프로세스로 전환

❑ 프로세스와 프로그램의 관계

- 프로그램이 프로세스로 변환
  - 운영체제로부터 프로세스 제어 블록을 얻는다는 뜻

- 프로세스 종료
  - 해당 프로세스 제어 블록이 폐기된다는 뜻

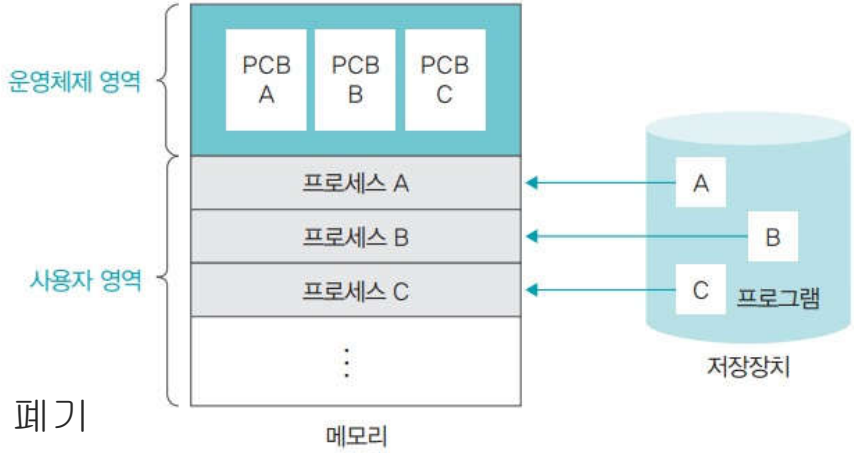
프로세스와 프로그램의 관계

- 프로세스 = 프로그램 + **PCB** + 사용자 데이터
- 프로그램 = 프로세스 – **PCB** – 사용자 데이터

4. 프로세스 제어 블록 (1/3)

❑ PCB 정의 및 역할

- 운영체제가 각 프로세스 관리하기 위해 정의한 자료 구조
  - 프로세스 구분자(PID) : 각 프로세스를 구분하는 구분자
  - 메모리 관련 정보 : 프로세스의 메모리 위치 정보
  - 각종 중간값 : 프로세스가 사용했던 중간값(문맥 교환 때 백업한 자료들)
- 각 프로세스는 고유의 PCB를 가짐
- 프로세스 생성 시 운영체제에 의해 만들어지고 프로세스 종료 시 폐기



[메모리내 PCB와 프로세스의 위치]

포인터	프로세스 상태
프로세스 구분자	
프로그램 카운터	
프로세스 우선순위	
각종 레지스터 정보	
메모리 관리 정보	
할당된 자원 정보	
계정 정보	
PPID와 CPID	
⋮	

[PCB의 구성]

4. 프로세스 제어 블록 (2/3)

❑ 구성 (1/2)

- **포인터 :**
  - 대기 상태에 있을 경우 입출력 요구 대상 프로세스끼리 **열결리스트로 관리**
- **프로세스 상태 :**
  - **생성, 준비, 실행, 대기,보류 상태** 등을 저장
- **프로세스 구분자(식별자) :**
  - 프로세스를 구분하기 위한 ID
- **프로그램 카운터 :**
  - 다음에 실행될 명령어의 위치값
- **프로세스 우선순위 :**
  - 실행 순서를 결정을 위한 우선순위
- **각종 레지스터 정보 :**
  - 실행되는 중에 사용하던 레지스터 값

포인터	프로세스 상태
프로세스 구분자	
프로그램 카운터	
프로세스 우선순위	
각종 레지스터 정보	
메모리 관리 정보	
할당된 자원 정보	
계정 정보	
PPID와 CPID	
⋮	

[PCB의 구성]

4. 프로세스 제어 블록 (3/3)

❑ 구성 (2/2)

- **메모리 관리 정보 :**
  - 프로세스의 메모리 위치 정보
  - 메모리 보호를 위해 사용하는 경계 레지스터 값과 한계 레지스터 값 등
- **할당된 자원 정보 :**
  - 입출력 자원이나 오픈 파일 등에 대한 정보
- **계정 정보 :**
  - 계정 번호, CPU 할당 시간, CPU 사용 시간 등
- **PPID와 CPID :**
  - PPID: 부모 프로세스 식별자
  - CPID: 자식 프로세스 식별자

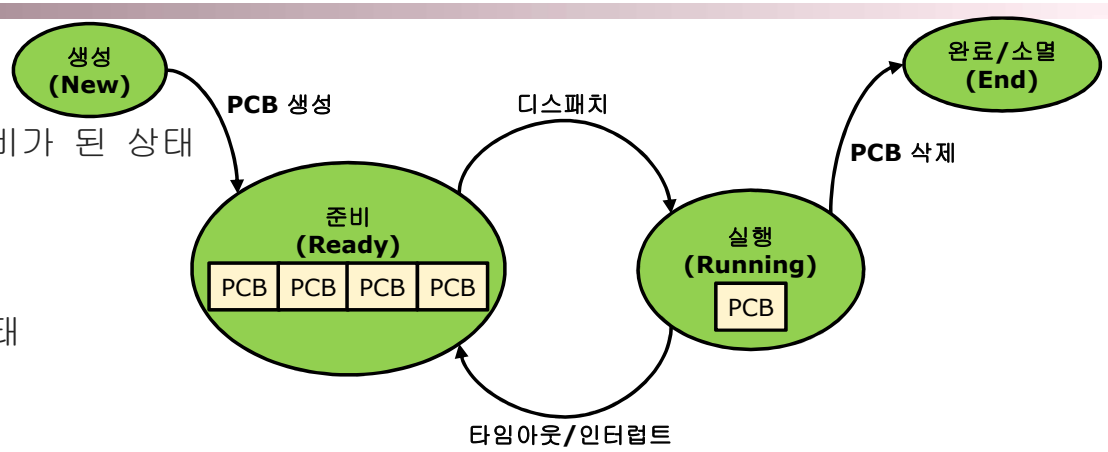
포인터	프로세스 상태
프로세스 구분자	
프로그램 카운터	
프로세스 우선순위	
각종 레지스터 정보	
메모리 관리 정보	
할당된 자원 정보	
계정 정보	
PPID와 CPID	
⋮	

[PCB의 구성]

5. 프로세스의 상태 (1/9)

❑ 네 가지 상태

- **생성 상태(New)** : 메모리에 적재되어 프로세스로 변환될 준비가 된 상태
- **준비 상태(Ready)** : CPU 점유를 위해 기다리는 상태
- **실행 상태(Running)** : CPU를 점유한 상태
- **완료 상태(End/Terminate)** : 프로세스 제어 블록이 사라진 상태



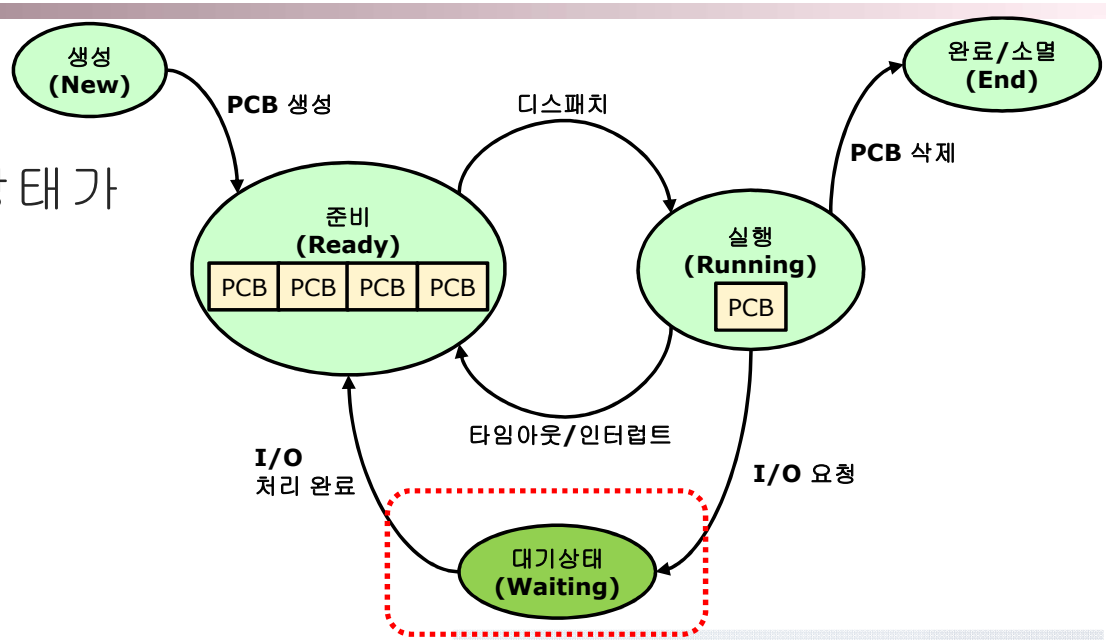
❑ 두 가지 행위

- **디스패치(Dispatch)** : 대기 상태에서 실행 상태로 변화는 과정
- **타임아웃(Timeout) 또는 인터럽트(Interrupt)** :
  실행상태에서 준비 상태로 진입을 위한 이벤트

5. 프로세스의 상태 (2/9)

❑ 다섯 가지 상태

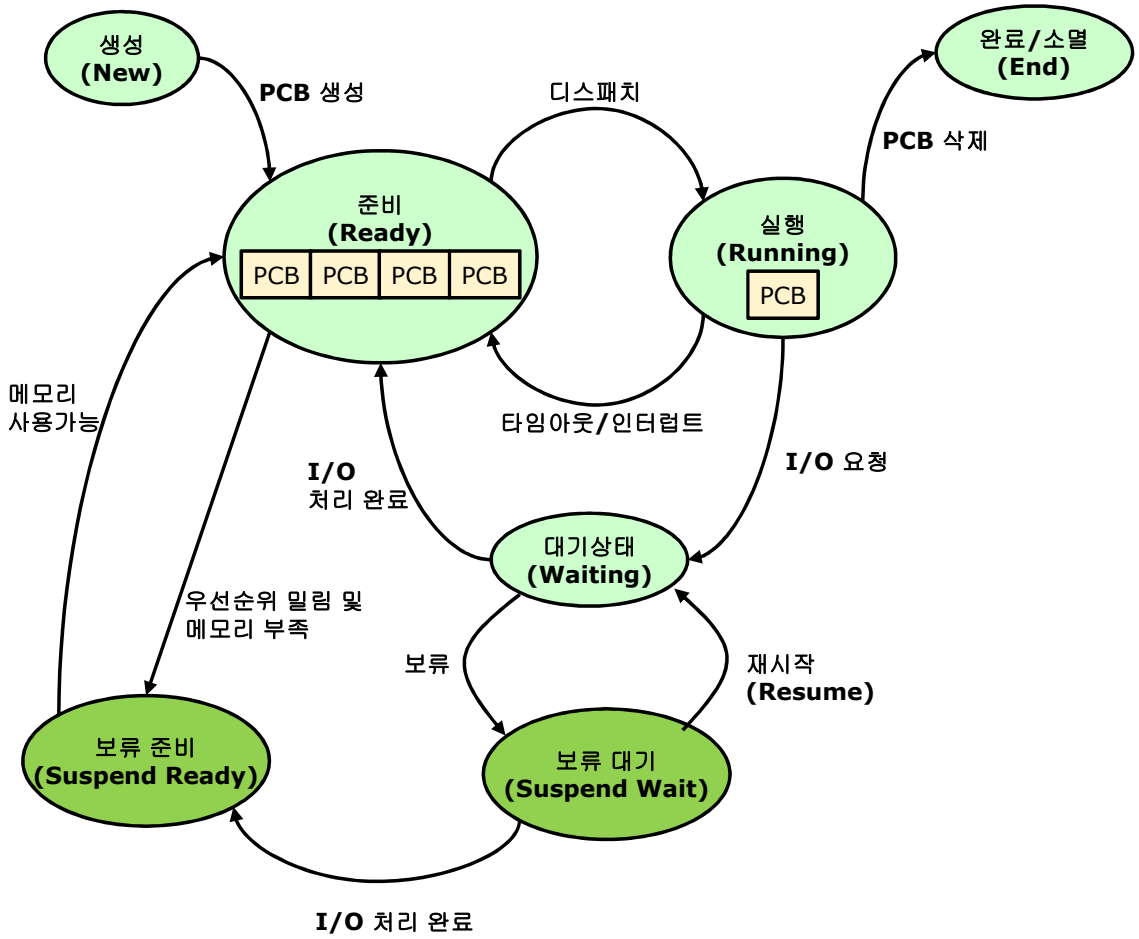
- 기존의 네 가지 상태에서 한가지의 상태가 추가된 구조
- 대기 상태(Waiting 또는 Blocking)가 추가됨



5. 프로세스의 상태 (3/9)

인덕대학교

❑보류 상태를 포함한 프로세스 상태

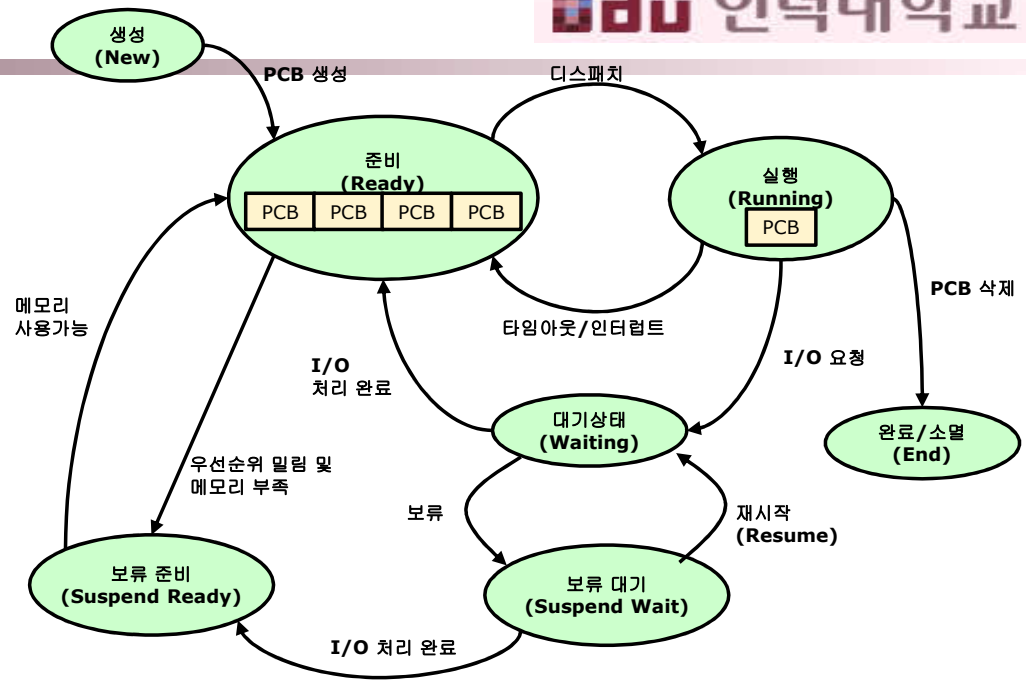


5. 프로세스의 상태 (4/9)

❑ 프로세스 상태 세부 (1/4)

- 생성 상태
  - 메모리에 적재되어 **프로세스로 변환될 준비가 된 상태**
  - 새로운 프로세스 제어 블록(PCB)이 운영체제에 등록됨
  - 생성된 프로세스는 준비 상태에서 자기 순서를 기다림

- 준비 상태
  - CPU를 점유하기 위해 대기 중인 상태
    - 대부분의 프로세스가 이 상태에 있음
  - PCB는 준비 큐에서 기다리며 CPU 스케줄러에 의해 관리



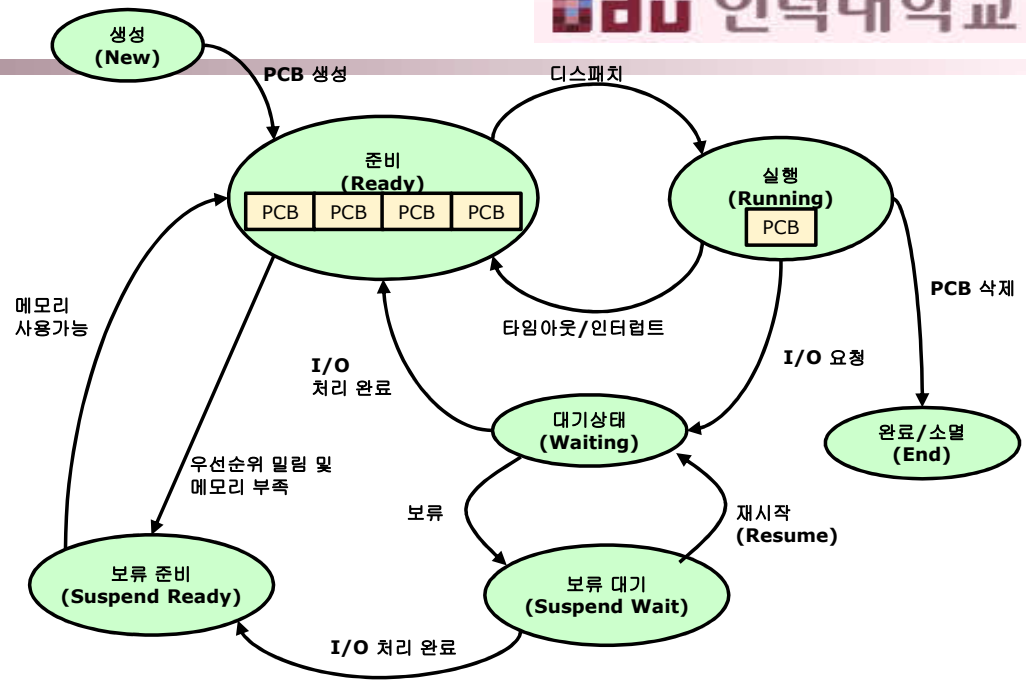
CPU 스케줄러 기능

- 준비 상태의 큐(queue) 관리
- 큐에 있는 어떤 프로세스의 PCB을 실행 상태로 보낼지 결정
- PCB의 선택에 의한 실행(CPU 점유)은 디스패치(dispatch)라고 함

5. 프로세스의 상태 (5/9)

❑ 프로세스 상태 세부 (2/4)

- 실행 상태
  - 프로세스가 CPU를 점유하고 있는 상태
  - 자신에게 주어진 시간(타임 슬라이스) 동안만 CPU를 점유
  - 타임아웃되면 **준비 상태**로 전환
  - 작업이 완료되면 프로세스 종료(소멸, PCB 삭제)
  - 입출력을 요청이 있으면 **대기상태**로 진입
    - 입출력이 완료되면 준비 상태로 진입

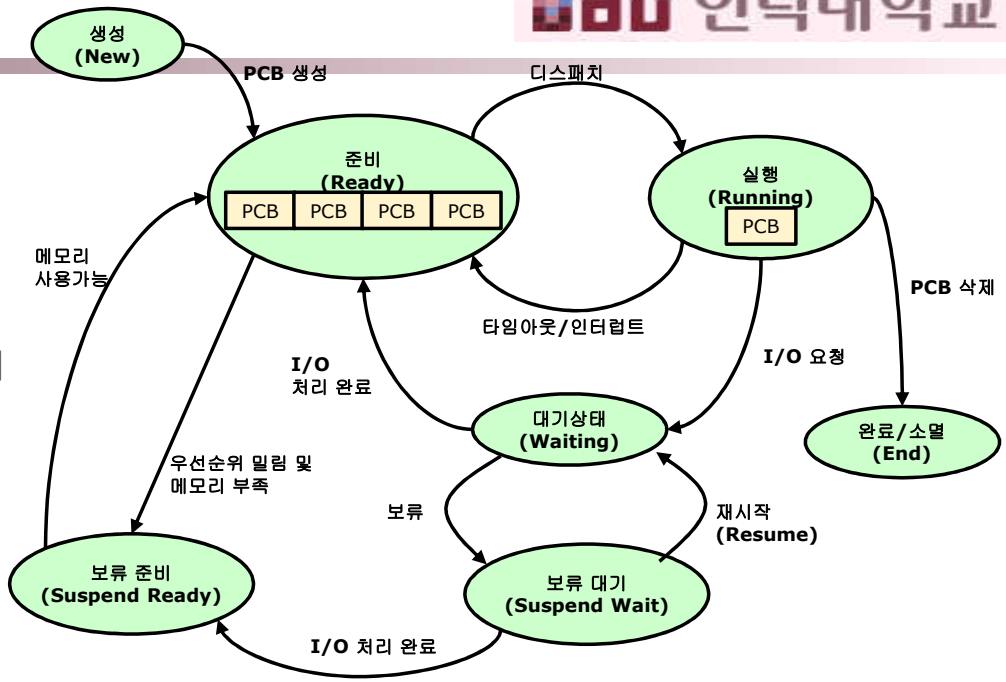


5. 프로세스의 상태 (6/9)

❑ 프로세스 상태 세부 (3/4)

- 대기 상태
  - 입출력이 완료될 때까지 기다리는 상태
  - 입출력 완료 인터럽트가 발생하면 입출력장치별 큐에서 빠져나와 준비 상태로 진입

- 완료 상태
  - 프로세스가 종료되는 상태
  - 메모리내 코드와 데이터를 메모리에서 삭제
  - PCB 폐기



코어 덤프(Core Dump)

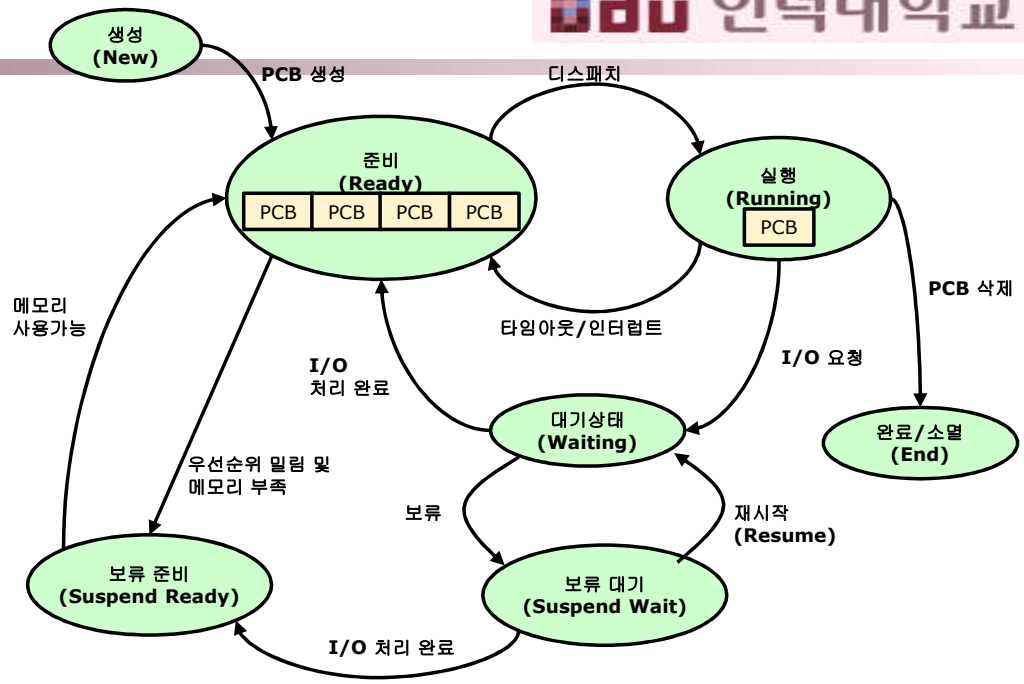
- 프로세스의 상태를 분석하기 위해 해당 프로세스의 메모리내 기록정보(상태정보 포함)를 파일로 저장하는 과정
- 주로 디버깅(Debugging)용도로 많이 활용

5. 프로세스의 상태 (7/9)

❑ 프로세스 상태 세부 (4/4)

▪ 보류 상태

- 프로세스가 준비 또는 대기 상태에서 **한동안 변화가 없을 때** 보류 상태로 전환됨
- 다음과 같은 경우에 보류 상태가 됨
  - 실행 메모리 **공간이 부족**할 때
  - 프로그램에 오류가 있어서 **실행을 계속 미루는 상황**
  - 바이러스와 같이 **악의적인 공격(잘못된 자원 접근을 시도)을 하는 프로세스라고 판단**될 때
  - **간헐적으로 실행**되는 프로세스로 우선순위에서 뒤에 있을 때
  - 입출력을 기다리는 프로세스의 **입출력이 계속 지연될 때**
  - 예: 웹브라우저에서 응답없음 대기상황



❑ 프로세스 기타 상태 (1/2)

- 좀비(Zombie) 상태
  - 운영체제가 프로세스의 소멸(폐기)를 못한 프로세스
  - 실행 상태 진입을 하지 않으며 메모리에 잔존

존비 발생의 대표적 사유

- 운영체제에서 관리되는 프로세스들은 부모와 자식 관계의 트리 형태로 관리됨
- 자식 프로세스의 종료는 부모 프로세스에게 결과(성공, 실패 등)를 알림
- 부모 프로세스는 자식프로세스의 종료 상태를 통해 타 기능을 처리할 수 있음
- 부모 프로세스가 자식프로세스의 결과를 읽지 않을 경우, 자식 프로세스는 종료 함수(exit(int)) 호출 후 무한 대기상태가 됨

❑ 프로세스 기타 상태 (2/2)

- 정지(휴식) 상태
  - 프로세스가 작업을 일시적으로 쉬고 있는 상태
  - 유닉스 쉘에서 프로그램 중에 [Ctrl + Z]를 누르면 볼 수 있는 상태

```
[kali@kali ~]$sleep 100
^Z
[1]+  Stopped                 sleep 100
[kali@kali ~]$jobs
[1]+  Stopped                 sleep 100
[kali@kali ~]$bg
[1]+ sleep 100 &
[kali@kali ~]$jobs
[1]+  Running                 sleep 100 &
[kali@kali ~]$
```

수고하셨습니다.