

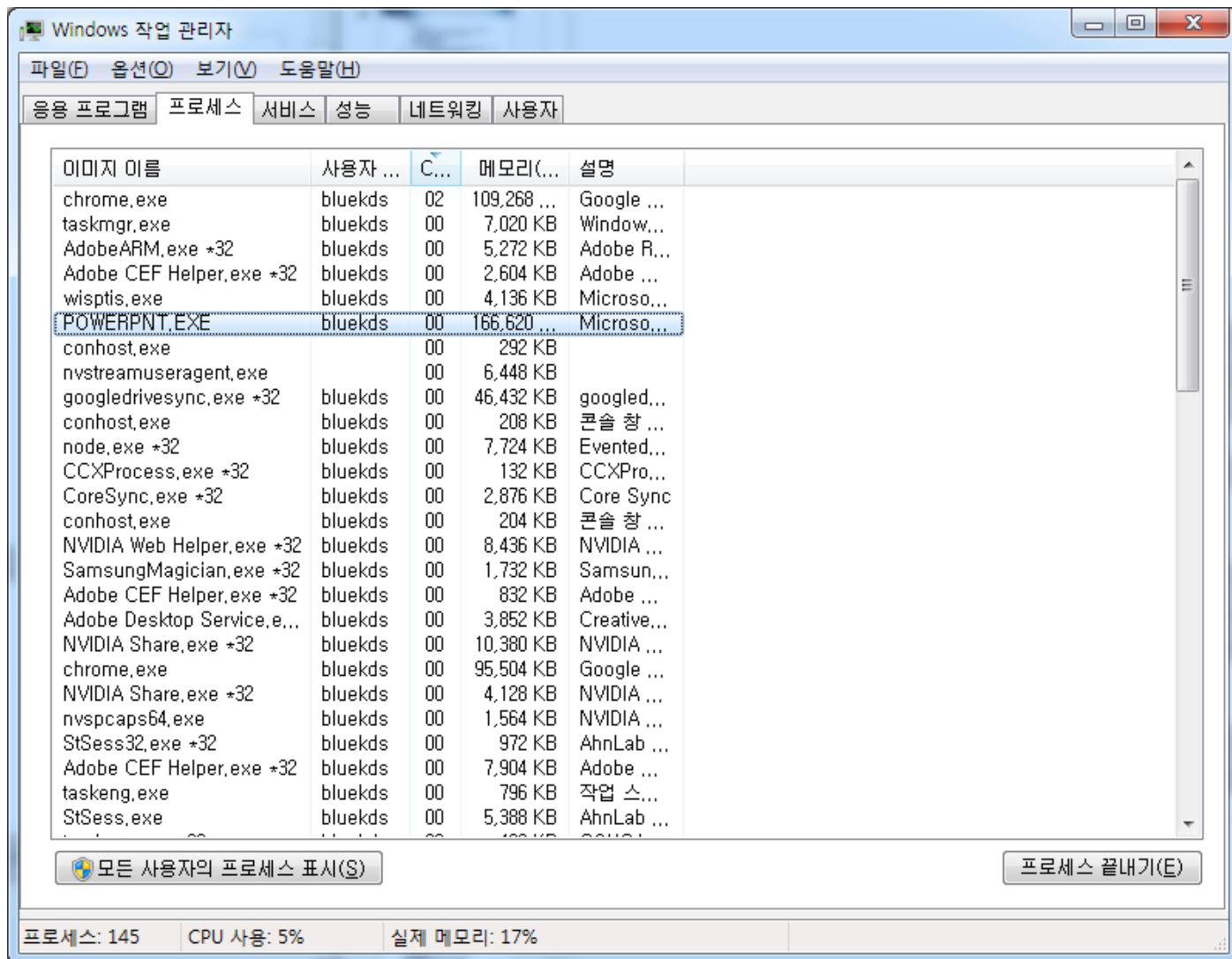
Chapter 3-1

프로세스 관리

Process Management



프로세스 (Process)

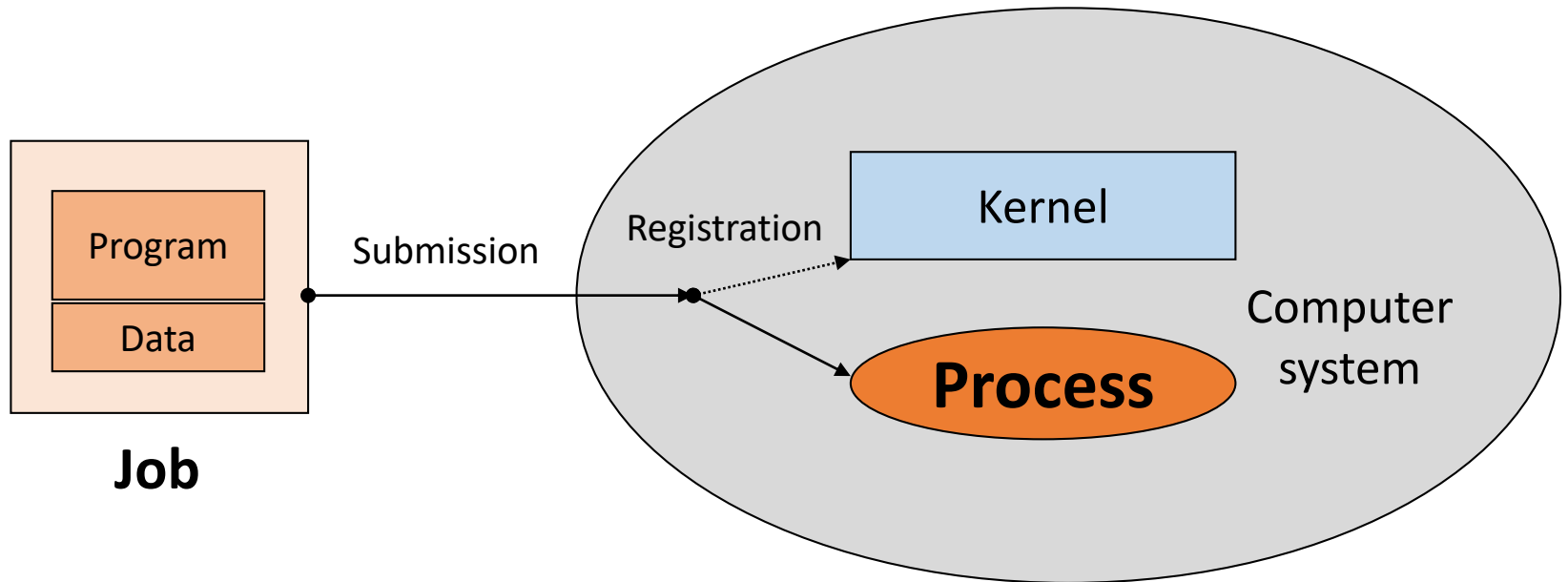


Job vs Process

- **작업 (Job) / 프로그램 (Program)**
 - 실행 할 프로그램 + 데이터
 - 컴퓨터 시스템에 실행 요청 전의 상태
- **프로세스 (Process)**
 - 실행을 위해 시스템(커널)에 등록된 작업
 - 시스템 성능 향상을 위해 커널에 의해 관리 됨



Job vs Process



Base images from Prof. Seo's slides



Job vs Process

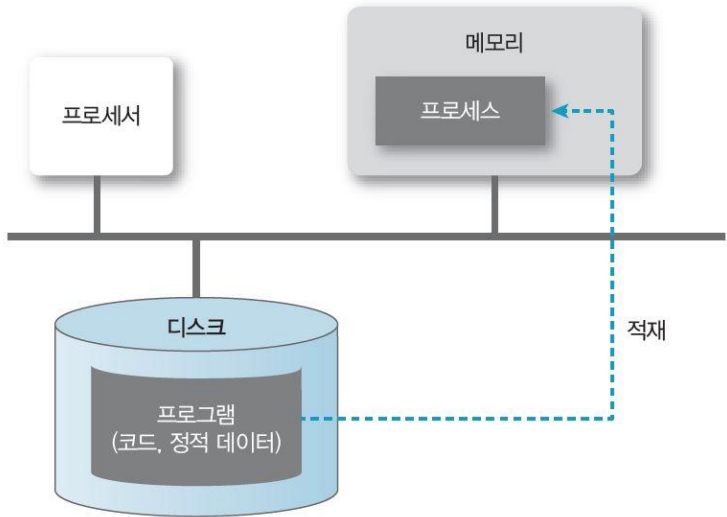


그림 3-1 프로그램과 프로세스 : 프로그램이 메모리로 적재되면 프로세스가 됨

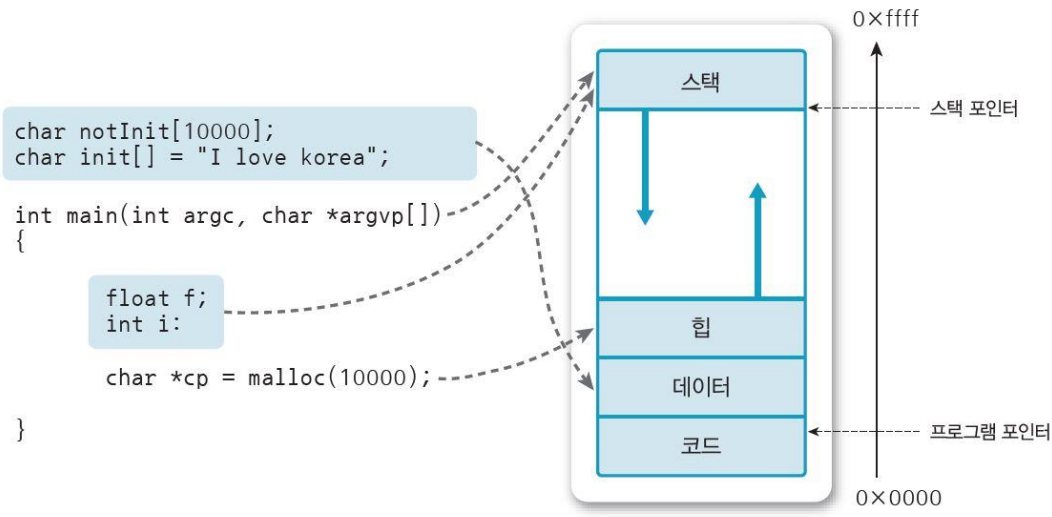


그림 3-2 프로세스의 일반적인 메모리 구조(사용자 관점의 프로세스)



프로세스의 정의

- **실행중인 프로그램**

- 커널에 등록되고 커널의 관리하에 있는 작업
- 각종 자원들을 요청하고 할당 받을 수 있는 개체
- 프로세스 관리 블록(PCB)을 할당 받은 개체
- 능동적인 개체(active entity)
 - 실행 중에 각종 자원을 요구, 할당, 반납하며 진행

- **Process Control Block (PCB)**

- 커널 공간 (kernel space) 내에 존재
- 각 프로세스들에 대한 정보를 관리



프로세스의 종류

표 3-1 프로세스의 종류

| 구분 | 종류 | 설명 |
|----------|--------------|---|
| 역할 | 시스템(커널) 프로세스 | 모든 시스템 메모리와 프로세서의 명령에 액세스할 수 있는 프로세스이다. 프로세스 실행 순서를 제어하거나 다른 사용자 및 커널(운영체제) 영역을 침범하지 못하게 감시하고, 사용자 프로세스를 생성하는 기능을 한다. |
| | 사용자 프로세스 | 사용자 코드를 수행하는 프로세스이다. |
| 병행 수행 방법 | 독립 프로세스 | 다른 프로세스에 영향을 주지 않거나 다른 프로세스의 영향을 받지 않으면서 수행하는 병행 프로세스이다. |
| | 협력 프로세스 | 다른 프로세스에 영향을 주거나 다른 프로세스에서 영향을 받는 병행 프로세스이다. |

Images from the text book



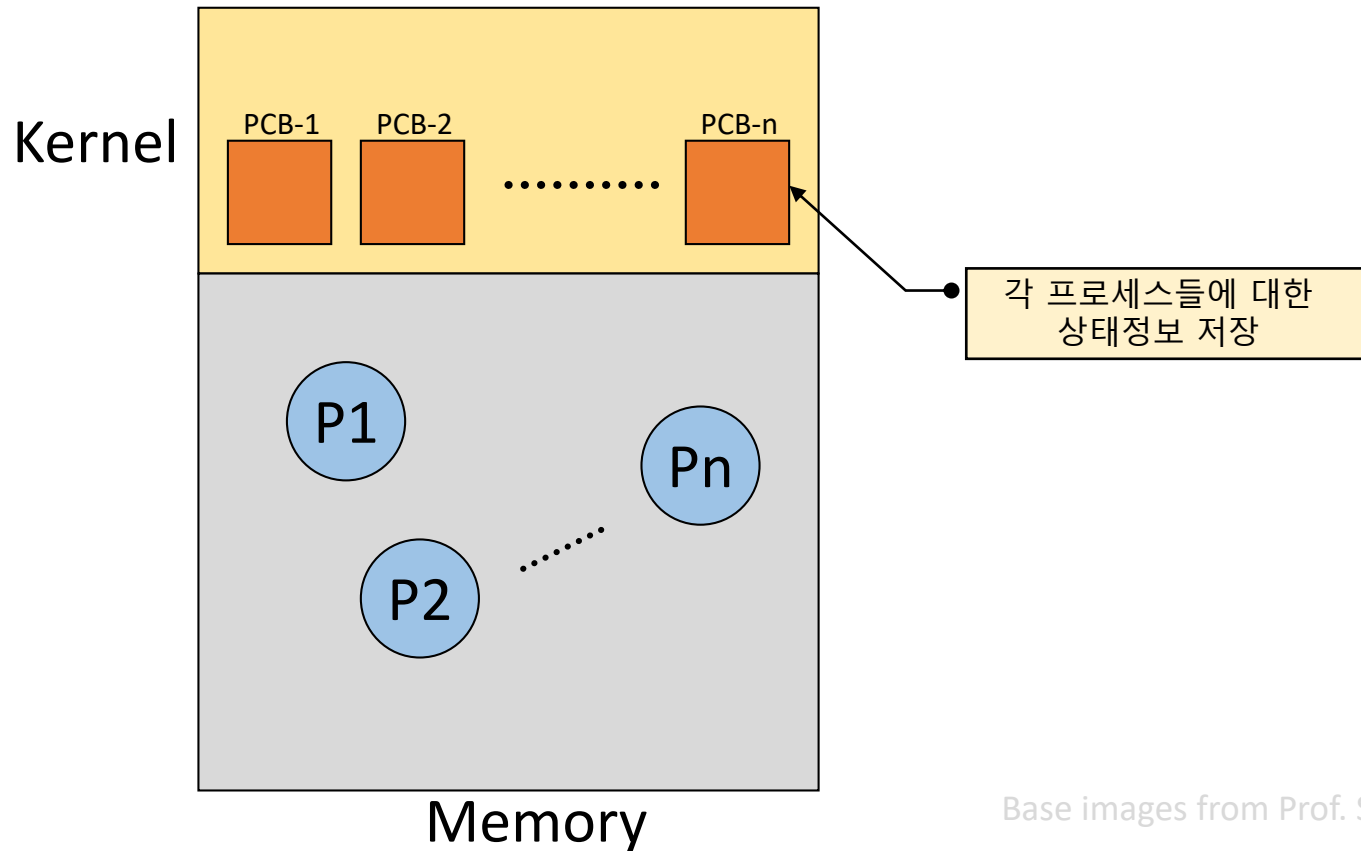
자원(Resource)의 개념

- 커널의 관리 하에 프로세스에게 할당/반납 되는 수동적 개체(passive entity)
- 자원의 분류
 - H/W resources
 - Processor, memory, disk, monitor, keyboard, Etc.
 - S/W resources
 - Message, signal, files, installed SWs, Etc.



Process Control Block (PCB)

- OS가 프로세스 관리에 필요한 정보 저장
- 프로세스 생성 시, 생성 됨



Base images from Prof. Seo's slides



PCB가 관리하는 정보

- **PID : Process Identification Number**
 - 프로세스 고유 식별 번호
- **스케줄링 정보**
 - 프로세스 우선순위 등과 같은 스케줄링 관련 정보들
- **프로세스 상태**
 - 자원 할당, 요청 정보 등
- **메모리 관리 정보**
 - Page table, segment table 등
- **입출력 상태 정보**
 - 할당 받은 입출력 장치, 파일 등에 대한 정보 등
- **문맥 저장 영역 (context save area)**
 - 프로세스의 레지스터 상태를 저장하는 공간 등
- **계정 정보**
 - 자원 사용 시간 등을 관리

- PCB 정보는 OS 별로 서로 다름
- PCB 참조 및 갱신 속도는 OS의 성능을 결정 짓는 중요한 요소 중 하나



개요

- 프로세스의 개념
- PCB (프로세스 관리 블록)
- 프로세스 상태 변화
- 인터럽트
- Context switching



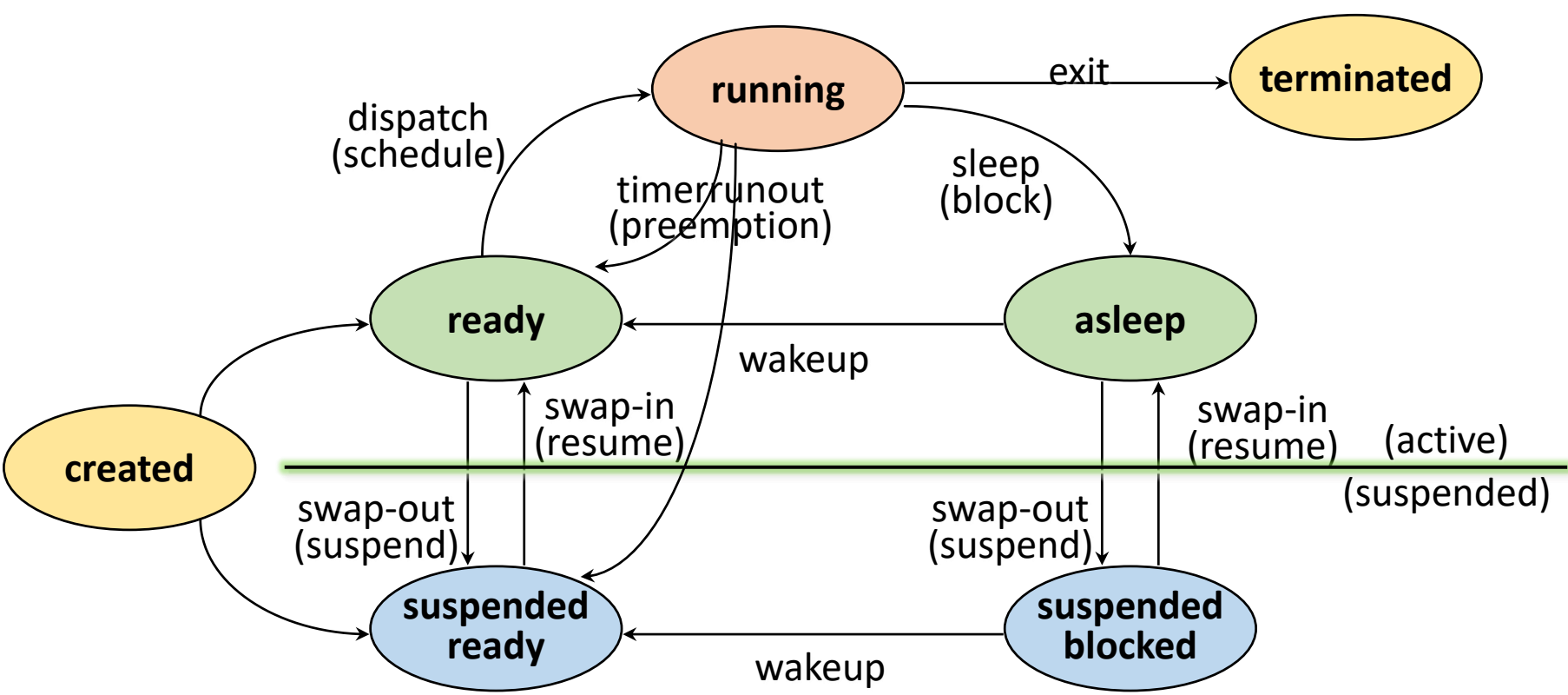
프로세스의 상태 (Process States)

- 프로세스 - 자원 간의 상호작용에 의해 결정
- 프로세스 상태 및 특성

| 상태 | | 자원 할당 상태 | |
|----------------------------|-----------------|-----------------|-------|
| Active (swapped-in) | Running | 프로세서 0 | 메모리 0 |
| | Ready | 프로세서 x, 기타 자원 0 | |
| | Blocked, asleep | 프로세서 x, 기타 자원 x | |
| Suspended (Swapped-out) | Suspended ready | 프로세서 x | 메모리 x |
| | Suspended block | 프로세서 x, 기타 자원 x | |



Process State Transition Diagram

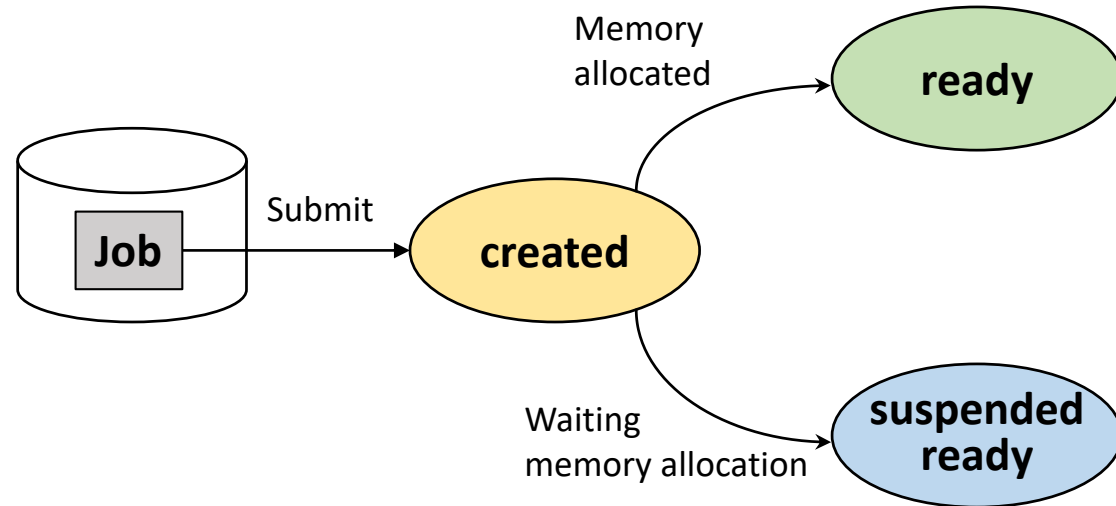


Base images from Prof. Seo's slides



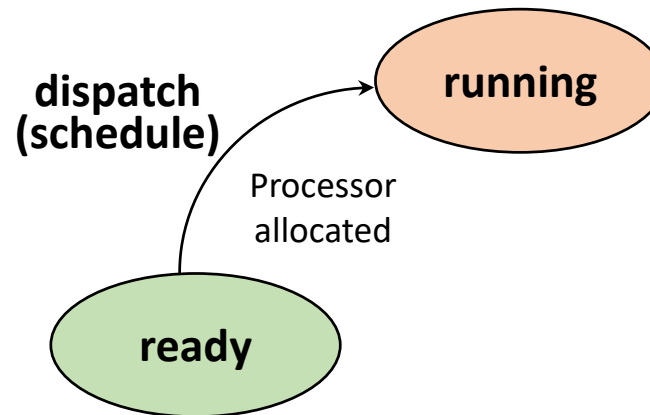
Created State

- 작업(Job)을 커널에 등록
- PCB 할당 및 프로세스 생성
- 커널
 - 가용 메모리 공간 체크 및 프로세스 상태 전이
 - Ready or Suspended ready



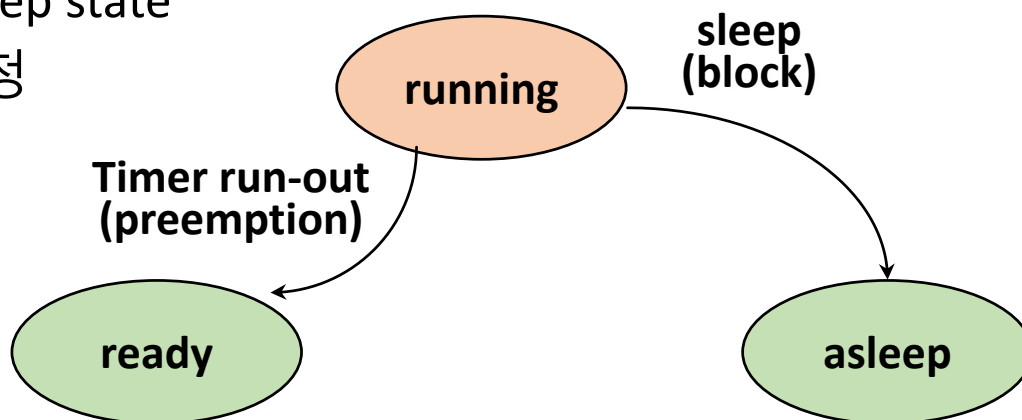
Ready State

- 프로세서 외에 다른 모든 자원을 할당 받은 상태
 - 프로세서 할당 대기 상태
 - 즉시 실행 가능 상태
- **Dispatch (or Schedule)**
 - Ready state → running state



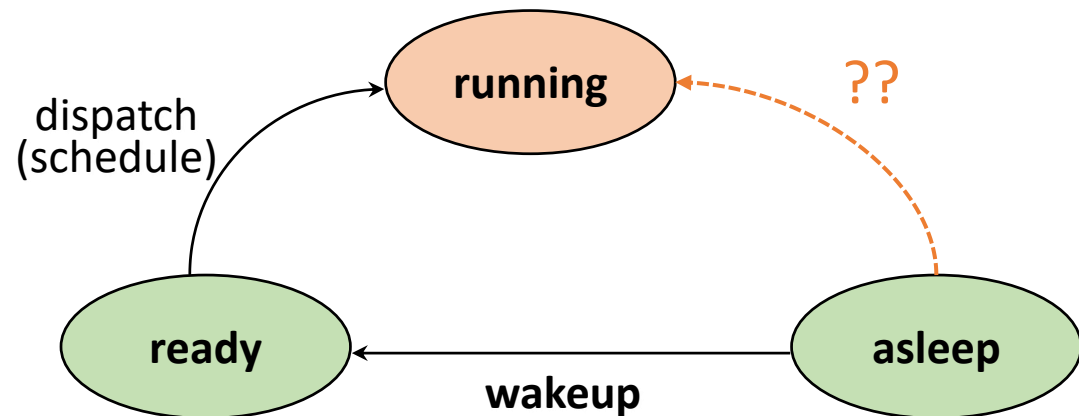
Running State

- 프로세서와 필요한 자원을 모두 할당 받은 상태
- **Preemption**
 - Running state → ready states
 - 프로세서 스케줄링 (e.g, time-out, priority changes)
- **Block/sleep**
 - Running state → asleep state
 - I/O 등 자원 할당 요청



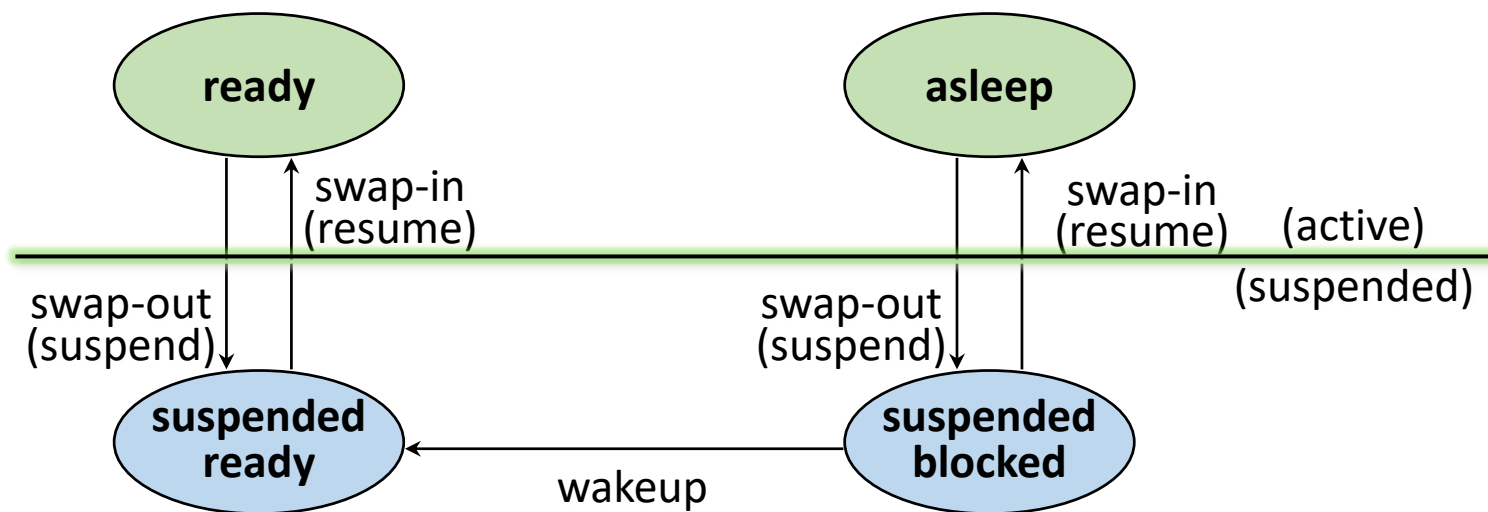
Blocked/Asleep State

- 프로세서 외에 다른 자원을 기다리는 상태
 - 자원 할당은 System call에 의해 이루어 짐
- **Wake-up**
 - Asleep state → ready state



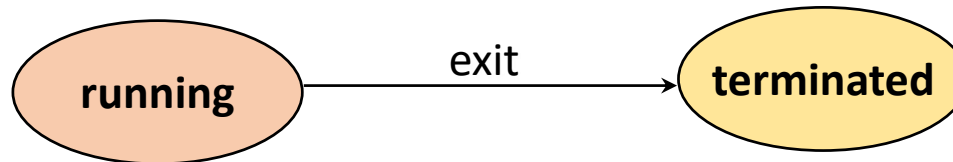
Suspended State

- 메모리를 할당 받지 못한(빠앗긴) 상태
 - Memory image를 swap device에 보관
 - Swap device: 프로그램 정보 저장을 위한 특별한 파일 시스템
 - 커널 또는 사용자에 의해 발생
- Swap-out(suspended), Swap-in(resume)



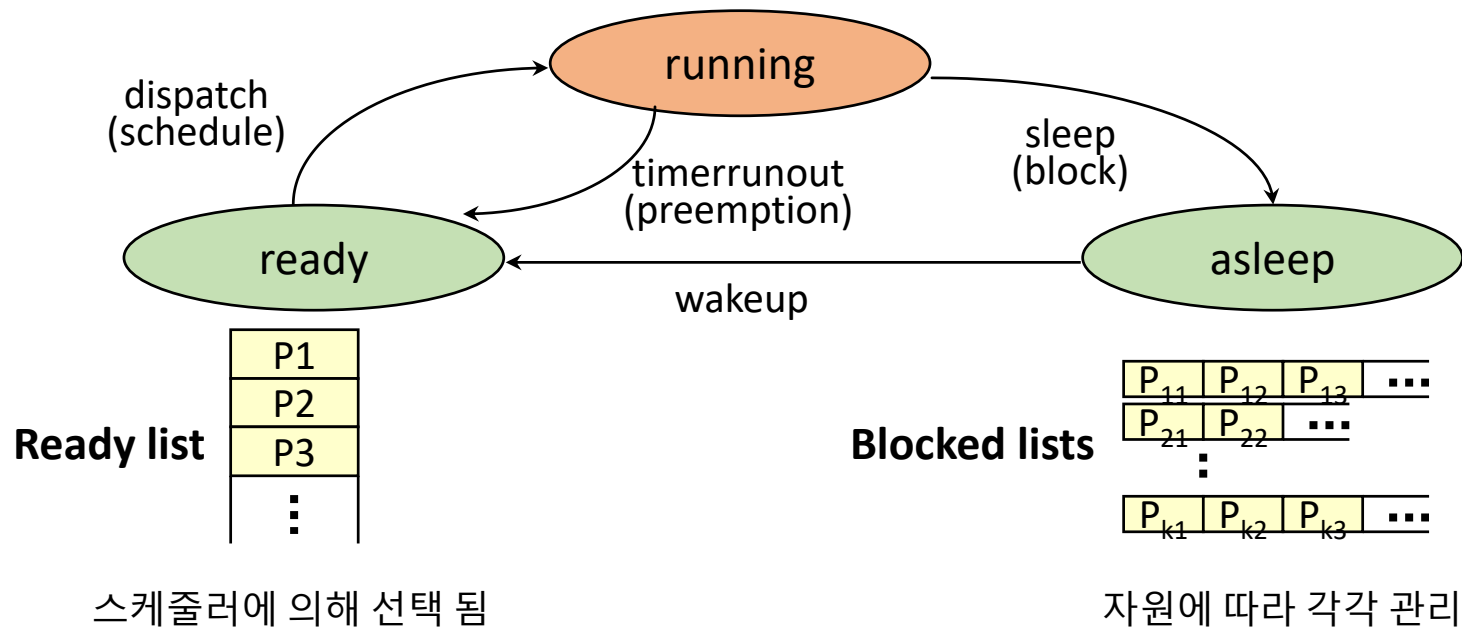
Terminated/Zombie State

- 프로세스 수행이 끝난 상태
- 모든 자원 반납 후,
- 커널 내에 일부 PCB 정보만 남아 있는 상태
 - 이후 프로세스 관리를 위해 정보 수집



프로세스 관리를 위한 자료구조

- Ready Queue
- I/O Queue
- Device Queue



개요

- 프로세스의 개념
- PCB (프로세스 관리 블록)
- 프로세스 상태 변화
- 인터럽트
- **Context switching**

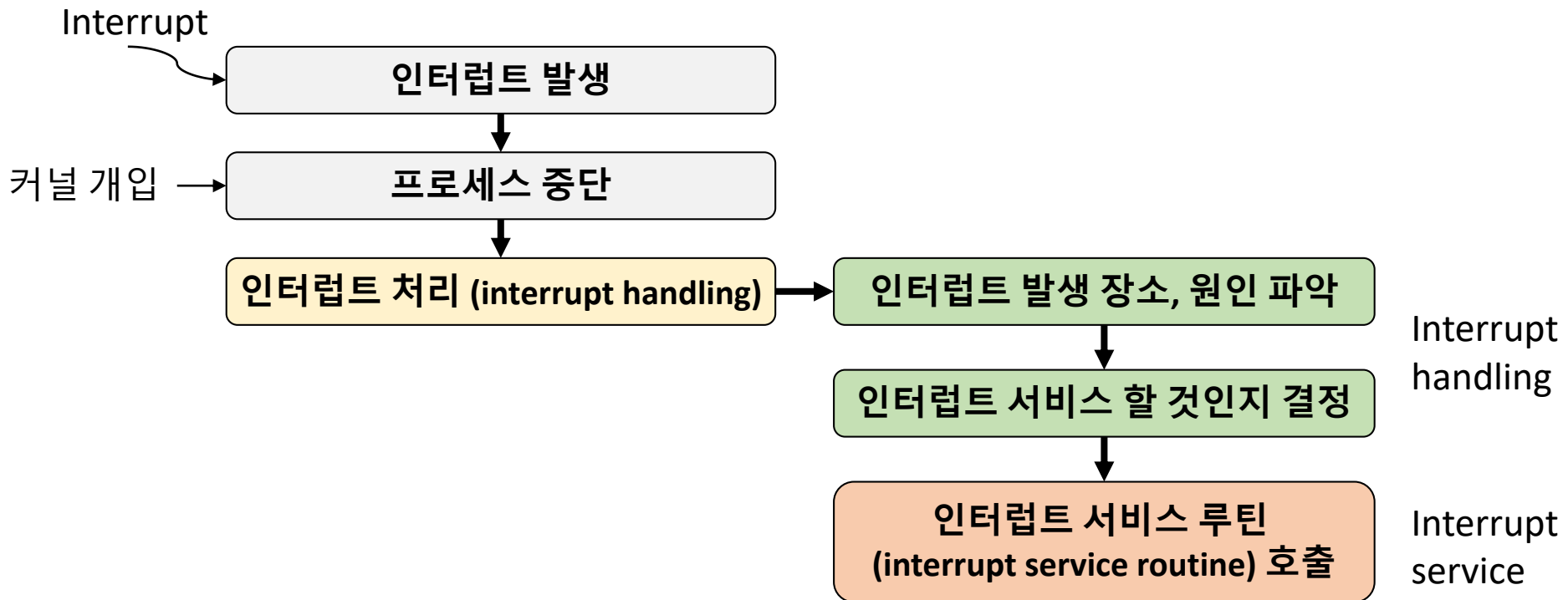


인터럽트(Interrupt)

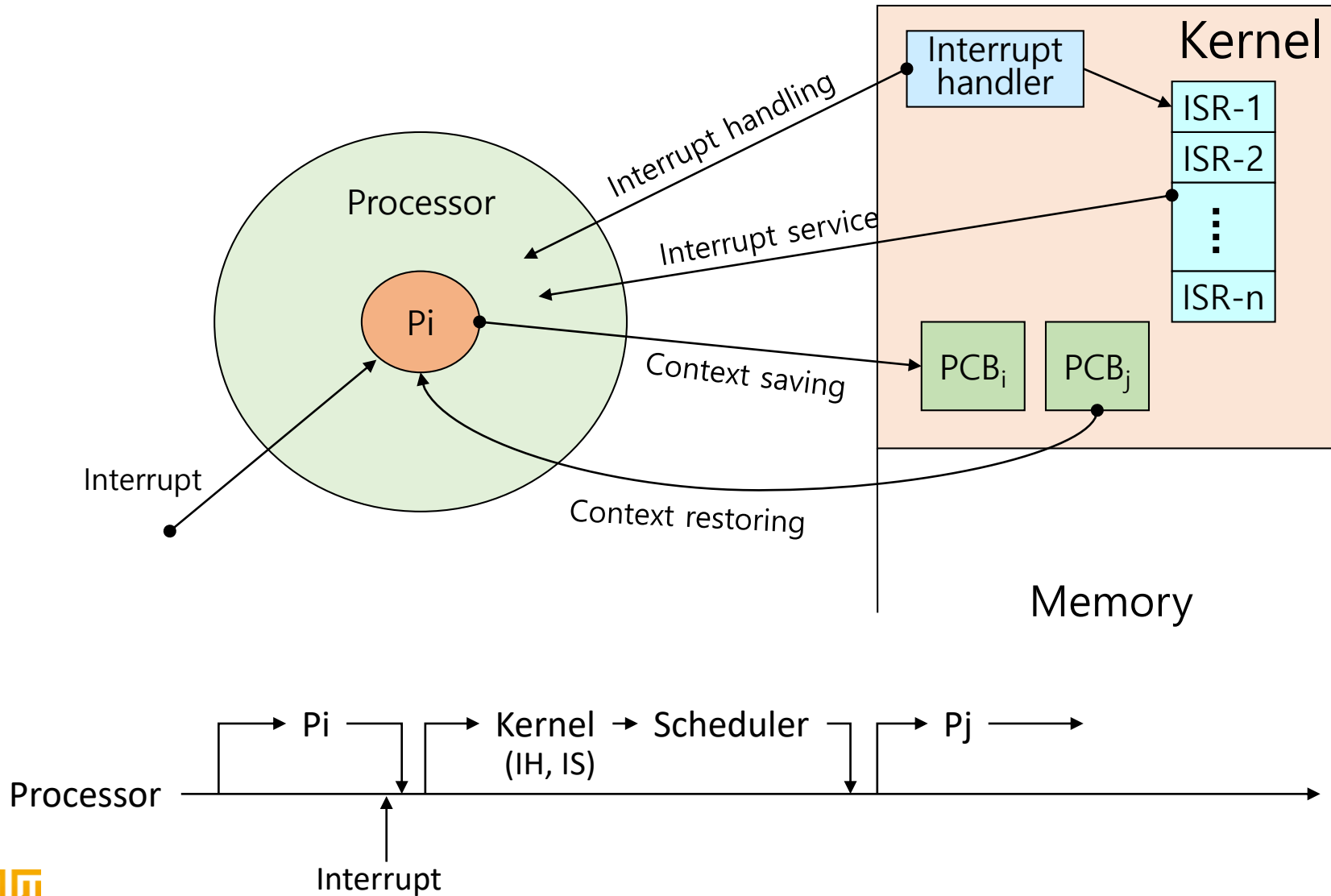
- 예상치 못한, 외부에서 발생한 이벤트
 - Unexpected, external events
- 인터럽트의 종류
 - I/O interrupt
 - Clock interrupt
 - Console interrupt
 - Program check interrupt
 - Machine check interrupt
 - Inter-process interrupt
 - System call interrupt



인터럽트 처리 과정



인터럽트 처리 과정



Context Switching (문맥 교환)

- **Context**

- 프로세스와 관련된 정보들의 집합
 - CPU register context => in CPU
 - Code & data, Stack, PCB => in memory

- **Context saving**

- 현재 프로세스의 Register context를 저장하는 작업

- **Context restoring**

- Register context를 프로세스로 복구하는 작업

- **Context switching** \cong Process switching

- 실행 중인 프로세스의 context를 저장하고, 앞으로 실행 할 프로세스의 context를 복구 하는 일
 - 커널의 개입으로 이루어짐



Context Switch Overhead

- **Context switching에 소요되는 비용**
 - OS마다 다름
 - OS의 성능에 큰 영향을 줌
- **불필요한 Context switching을 줄이는 것이 중요**
 - 예, 스레드(thread) 사용 등



요약

- 프로세스의 개념
- PCB (프로세스 관리 블록)
- 프로세스 상태 변화
- 인터럽트
- Context switching

