



# Python - HW8

임베디드스쿨1기

Lv1과정

2020. 10. 06

김인겸

# 1. Critical Section

process 1과 process2가 진행 중일 때  
두 개 이상의 프로세스가 동시 접근 가능한 구역을  
Critical section이라고 한다.

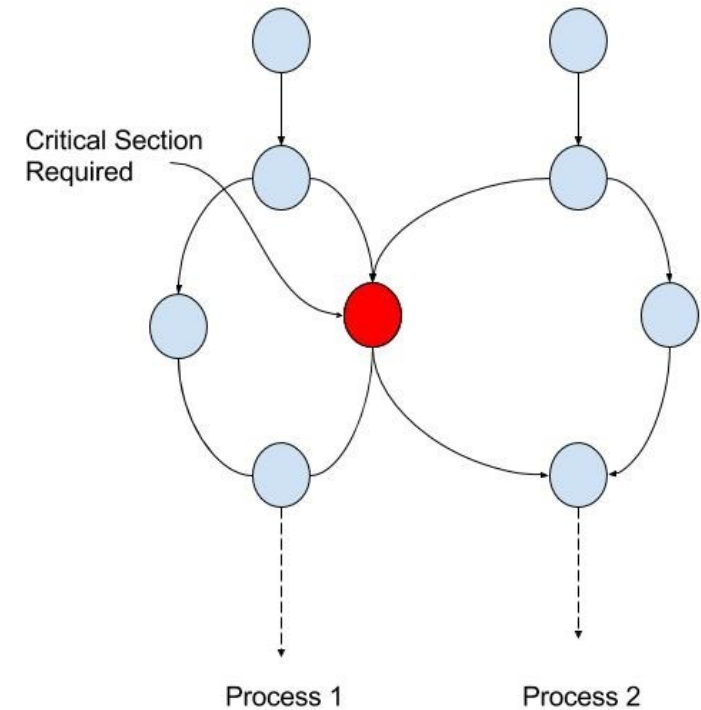
## Process A:

```
// Process A
.  
.  
b = x + 5;           // instruction executes at time = Tx  
.
```

## Process B:

```
// Process B  
.  
.  
x = 3 + z;           // instruction executes at time = Tx  
.
```

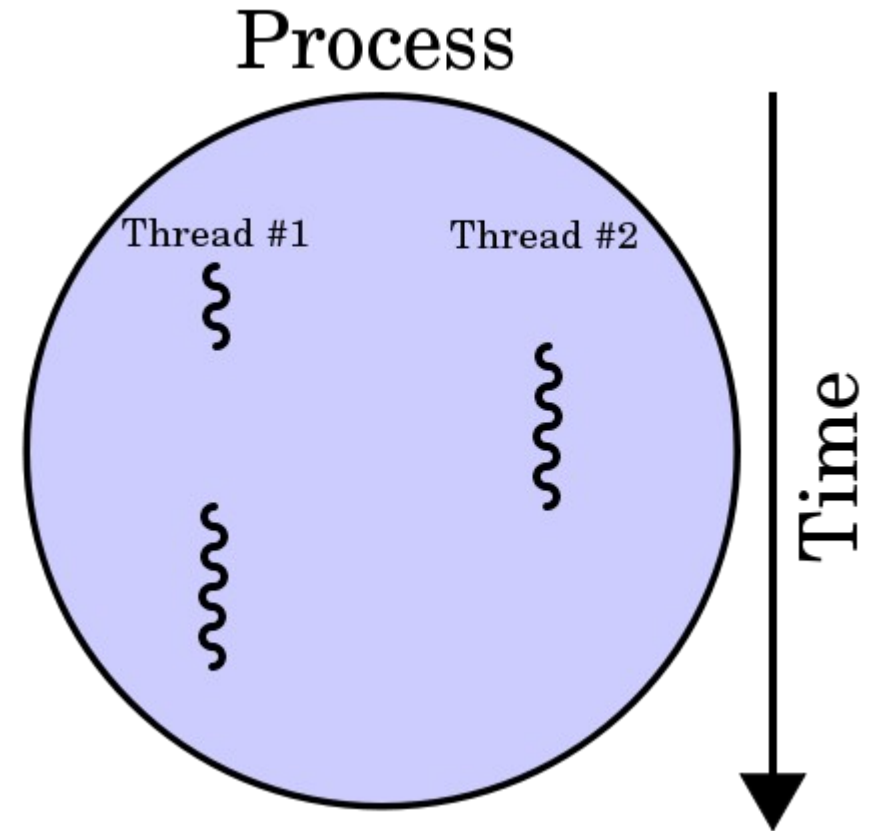
같은 시간 동안 두 개 이상의 프로세스가 공유 자원에 동시에 접근한다면  
원하지 않는 결과값을 얻을 수 있다  
이를 방지 하기 위해 세마포어, 뮉텍스같은 기술이 필요하다



## 2. Multi-Tasking

Task(태스크) : 작업의 최소 단위로서 프로세스가 될 수도 있고 멀티쓰레딩 환경에선 쓰레드가 될 수도 있다.

멀티태스킹(다중 작업) : 운영체제의 스케줄링을 통해 두 개 이상의 프로세스(혹은 태스크)가 실행될 시간을 분배받게 되고 프로세스 사이에 context switch가 발생하는데 이러한 병렬 연산을 멀티태스킹이라 한다,



# 3. IPC

---

-IPC(Inter Process Communication)

: 프로세스 간에 데이터 및 정보를 주고 받는 매커니즘을 말한다.

커널에서 IPC를 위한 도구를 제공하며, 시스템 콜의 형태로 프로세스에게 제공된다.

# 3. IPC 종류

## 1. PIPE(익명PIPE)

: 하나의 프로세스는 읽기만, 하나의 프로세스는 쓰기만 가능한 반이중 통신.

부모 자식 프로세스에서 주로 사용



## 2. Named PIPE

: 부모 프로세스와 무관하게 전혀 다른 모든 프로세스들 사이에서 통신 가능.

## 3. Message Queue

: 메모리를 이용한 PIPE

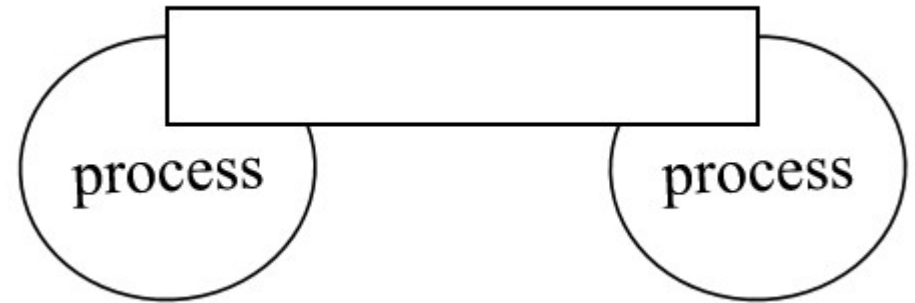
### 3. IPC 종류

#### 4. 공유 메모리(Shared Memory)

: 두 개 이상의 프로세서들이 주소 공간의 일부를 공유  
공유하는 방식으로 포인터를 이용한다.

커널 관여  $x \rightarrow$  IPC속도가 빠름.

구현 어려움

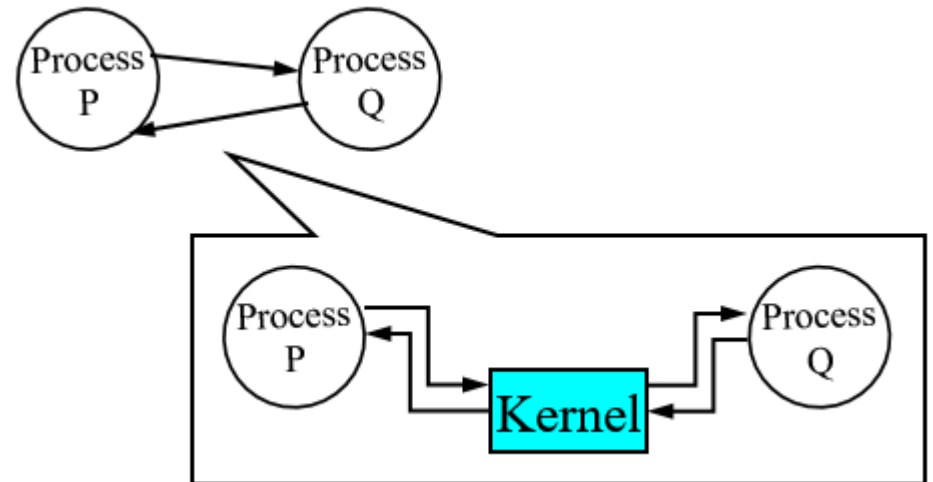


#### 5. 메시지 전달(Message Passing)

: 메모리 공유 없이 커널을 경유하여 고정길이/가변길이  
메시지를 주고받음.

커널이 관여하므로  $\rightarrow$  IPC속도 느림.

구현 간단



## 4. 우선순위

1. 비선점형 운영체제 : 현재 실행 중인 프로세스가 끝나야 다음 프로세스를 실행하는 것.

Process	Burst Time	Priority
$P_1$	10	3
$P_2$	1	1
$P_3$	2	4
$P_4$	1	5
$P_5$	5	2



평균 기다림 시간 : 8.2

우선 순위대로 프로세스를 실행하면 starving현상 발생.

starving현상이란? 우선순위가 낮으면 무한정 기다려야되는 현상.

해결방법 : 에이징(aging)

실행 중이지 않은 프로세스의 우선순위를 매분마다 1씩 증가시키는 방식

우선 순위가 낮은 프로세스가 실행중이지 않을 때 우선순위가 증가하므로 현재 진행 중인 프로세스보다 더 높은 우선순위를 갖는 역전현상이 발생하므로 starving현상을 해소할 수 있다

## 4. 우선순위

---

2. 선점형 운영체제 : 현재 실행 중인 프로세스보다 높은 우선순위를 갖는 프로세스가 등장하면 우선순위가 더 높은 프로세스를 실행하는 것.

- SRT(Shortest Remaining Time): 최단 잔여시간을 우선으로 하는 스케줄링.  
진행 중인 프로세스가 있어도, 최단 잔여시간인 프로세스를 위해 sleep시키고 짧은 프로세스를 먼저 할당한다.  
선점형 SJF 스케줄링이라 불린다.

RR(Round Robin) : Time Sharing System을 위해 설계된 스케줄링  
모든 프로세스가 같은 우선순위를 가지고, time slice를 기반으로 스케줄링한다.  
Time slice burst가 일어나면 해당 프로세스는 스케줄링 큐의 끝으로 이동한다.

<https://hyunah030.tistory.com/4>