



5강. 병행 프로세스 II

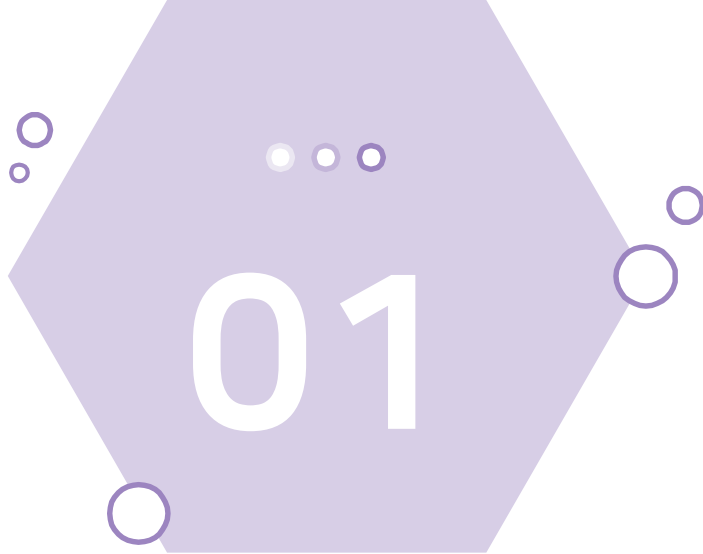
방송대 컴퓨터과학과
김진욱 교수



목차

01 프로세스의 상호협력

02 프로세스 간의 통신



프로세스의 상호협력

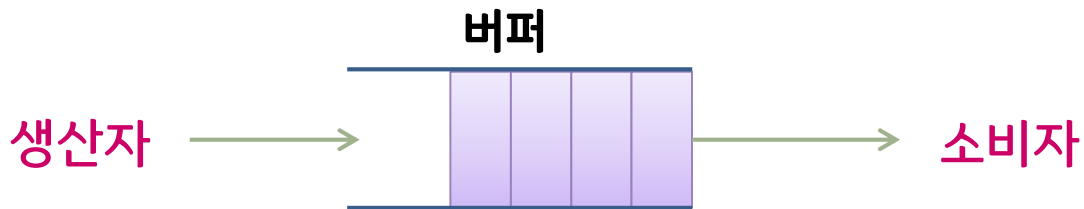
프로세스의 상호협력

■ 병행 프로세스들의 상호협력

- 공통작업을 수행하기 위해 서로 협동하는 경우
- 예: 생산자/소비자 문제, 판독기/기록기 문제

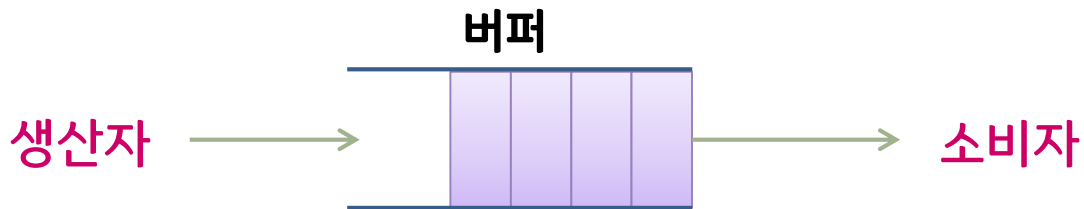
생산자/소비자 문제

■ 유한 버퍼 문제



생산자/소비자 문제

■ 유한 버퍼 문제



- 버퍼에 데이터를 채우는 프로세스

» 상호배제 필요

- 버퍼가 가득 차 있다면 생산자는 대기해야 함

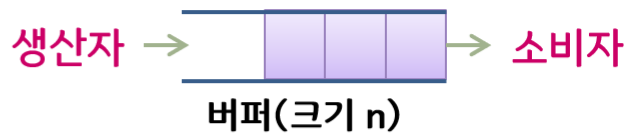
» 동기화 필요

- 버퍼에 있는 데이터를 읽어내는 프로세스

- 버퍼가 비어 있다면 소비자는 대기해야 함

생산자/소비자 문제

■ 세마포어 이용 – mutex, empty, full



생산자 프로세스

repeat

nextp에 데이터 항목을 생산

nextp를 버퍼에 넣음

until false;

소비자 프로세스

repeat

버퍼에서 데이터 항목을 꺼내
nextc에 넣음

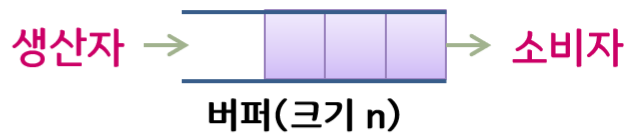
nextc를 소비

until false;

» 상호배제

생산자/소비자 문제

■ 세마포어 이용 – mutex, empty, full



생산자 프로세스

```
repeat
  nextp에 데이터 항목을 생산

  P(mutex);
  nextp를 버퍼에 넣음
  V(mutex);

until false;
```

소비자 프로세스

```
repeat

  P(mutex);
  버퍼에서 데이터 항목을 꺼내
  nextc에 넣음
  V(mutex);

  nextc를 소비

until false;
```

» 상호배제

- mutex의 초깃값 1

생산자/소비자 문제

■ 세마포어 이용 – mutex, empty, full



생산자 프로세스

```
repeat
    nextp에 데이터 항목을 생산
    P(empty);
    P(mutex);
    nextp를 버퍼에 넣음
    V(mutex);
    V(full);
until false;
```

소비자 프로세스

```
repeat
    P(full);
    P(mutex);
    버퍼에서 데이터 항목을 꺼내
    nextc에 넣음
    V(mutex);
    V(empty);
    nextc를 소비
until false;
```

» 상호배제

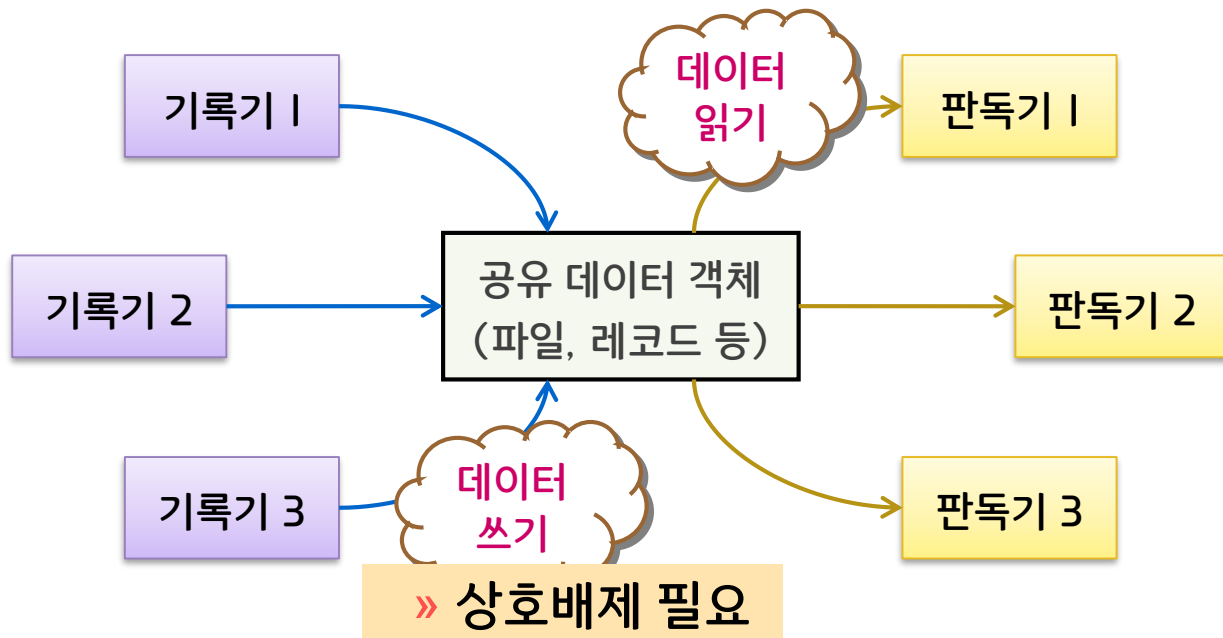
- mutex의 초깃값 1

» 동기화

- full의 초깃값 0
- empty의 초깃값 n

○ 판독기/기록기 문제

- 여러 개의 판독기가
동시에 공유 데이터 객체에 접근하는 것은 문제 없음



- 기록기가 다른 기록기 또는 판독기와
동시에 공유 데이터 객체에 접근하는 것은 문제

○ 판독기/기록기 문제

■ 우선순위에 따른 문제의 변형

• 제1 판독기/기록기 문제 (판독기 우선)

» 기록기가 이미 공유객체의 사용을 허가 받은 것이 아니라면
판독기는 대기하지 않음

→ 기록기의 기아상태 유발 가능

• 제2 판독기/기록기 문제 (기록기 우선)

» 일단 기록기가 준비되었다면
기록을 가능한 한 빨리 수행할 수 있도록 함

→ 판독기의 기아상태 유발 가능

제1 판독기/기록기 문제

■ 세마포어 이용 – mutex, wrt

기록기 프로세스

$P(wrt);$

쓰기 수행

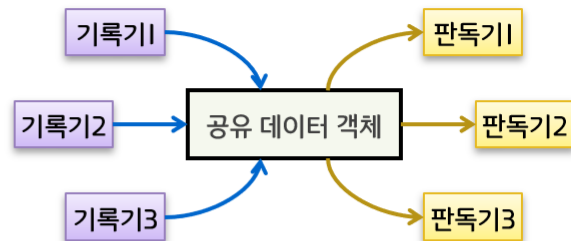
$V(wrt);$

판독기 프로세스

$P(wrt);$

읽기 수행

$V(wrt);$



» 상호배제

- wrt 의 초깃값 1

» 판독기 우선

제1 판독기/기록기 문제

■ 세마포어 이용 – mutex, wrt

기록기 프로세스

```
P(wrt);
쓰기 수행
V(wrt);
```

판독기 프로세스

```
P(mutex);

readcount:=readcount+1

if readcount=1 then P(wrt);

V(mutex);

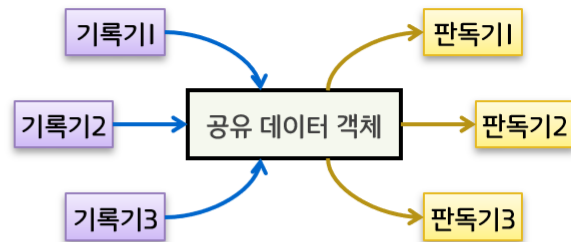
읽기 수행

P(mutex);

readcount:=readcount-1

if readcount=0 then V(wrt);

V(mutex);
```

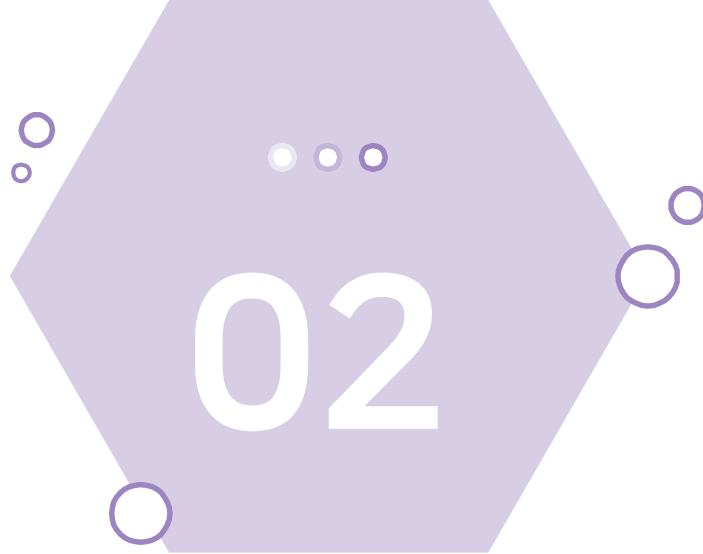


» 상호배제

- wrt의 초깃값 1

» 판독기 우선

- 변수 readcount의 초깃값 0
- mutex의 초깃값 1



프로세스 간의 통신

프로세스 간의 통신(IPC, InterProcess Communication)

- 병행 프로세스 사이의 통신을 위한 방법
 - 공유기억장치 기법
 - 메시지 시스템 기법
- 두 방법은 상호 배타적이지 않음
 - 단일 운영체제 내에서 동시에 사용 가능

프로세스 간의 통신

■ 공유기억장치 기법

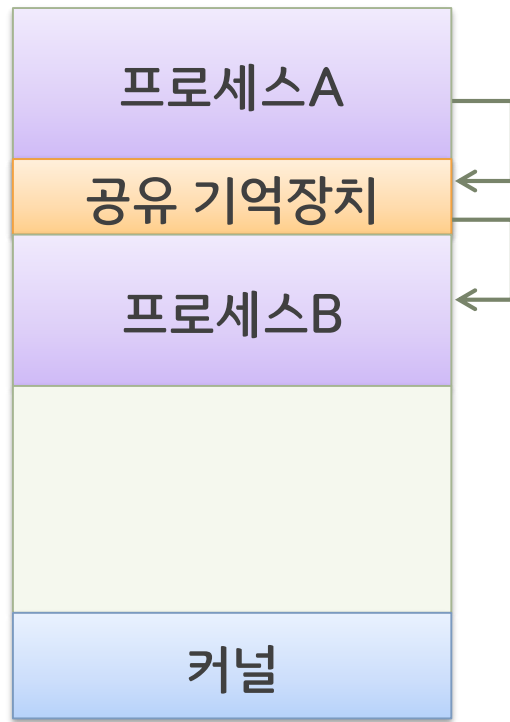
- 프로세스 간에 공유변수를 이용하여 정보를 교환



프로세스 간의 통신

■ 공유기억장치 기법

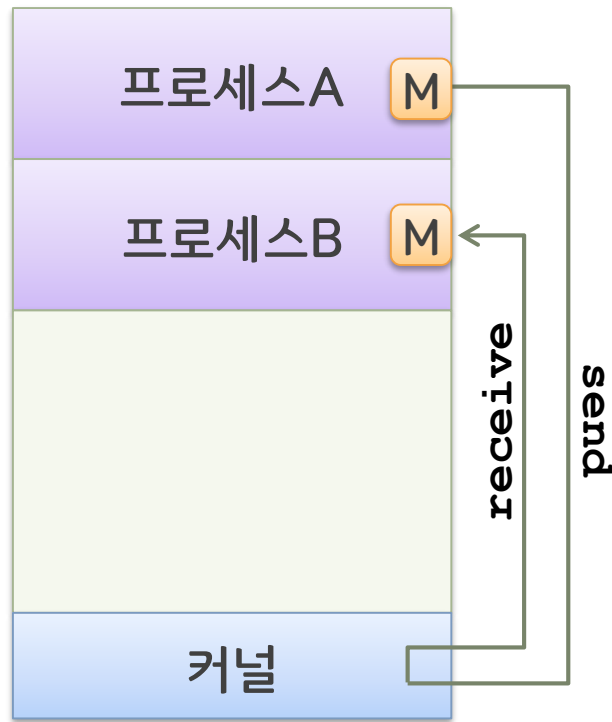
- 프로세스 간에 공유변수를 이용하여 정보를 교환
- 예: 유한 버퍼
- 고속 통신 가능
- 통신기능 제공의 책임: 응용 프로그래머



프로세스 간의 통신

■ 메시지 시스템 기법

- 메시지 교환방식으로 정보를 교환
- `send/receive` 연산자
- 소량의 데이터 교환에 유용
- 통신기능 제공의 책임: 운영체제



메시지 시스템 기법

■ 통신 링크

- 프로세스들 사이에 메시지를 주고받기 위한 연결통로



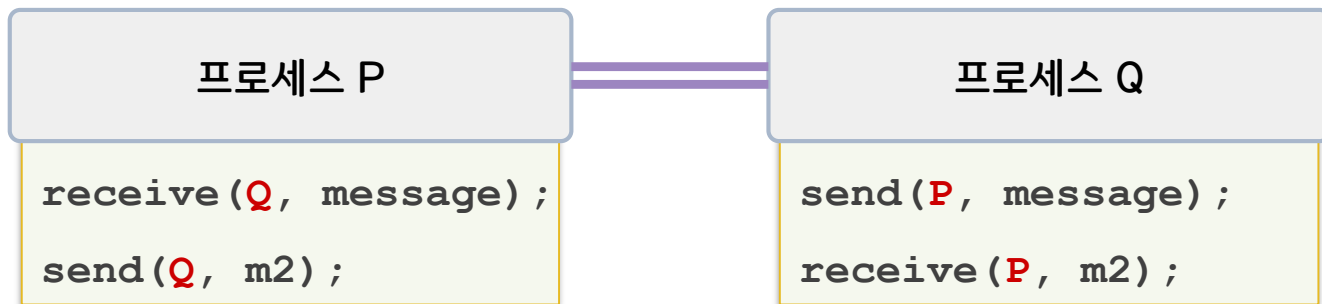
■ 논리적 구현에 대한 이슈

- 어떻게 링크를 설정?
- 한 링크가 2개 이상의 프로세스와 연결 가능?
두 프로세스 사이에 얼마나 많은 링크 존재?
- 링크의 용량은? 메시지의 크기는? 링크는 단방향인가 양방향인가?

메시지 시스템 기법

■ 직접 통신

- 메시지 전달 연산에 수신자나 송신자 이름을 명시

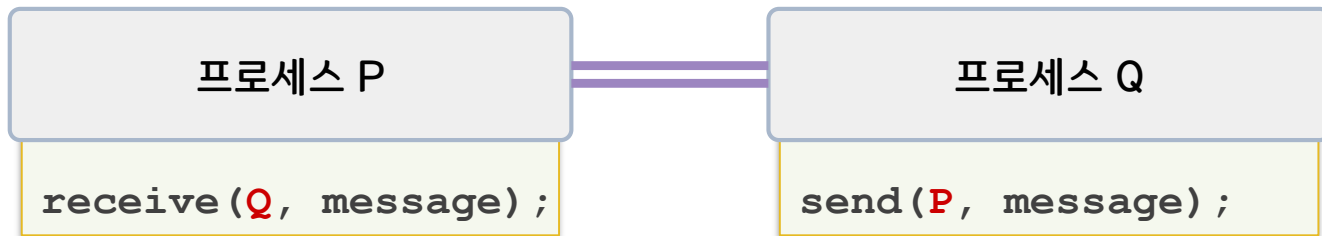


- 통신 링크는 자동 설정됨
- 하나의 링크는 두 프로세스 사이만 연관되며
각 통신 프로세스 쌍 사이에는 정확히 하나의 링크가 존재
- 링크는 양방향임

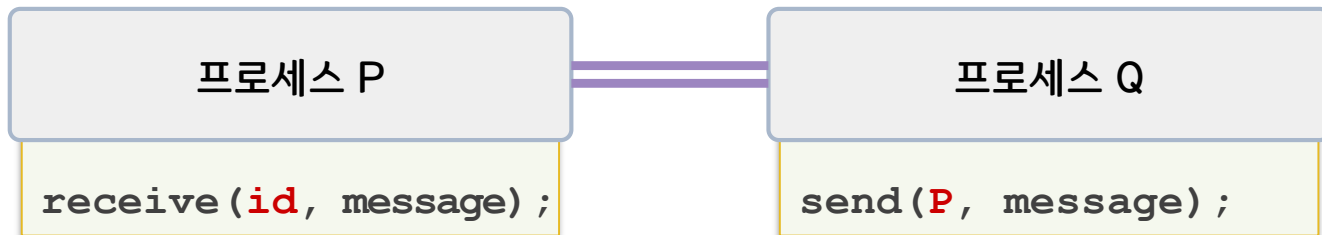
메시지 시스템 기법

■ 직접 통신

• 대칭형



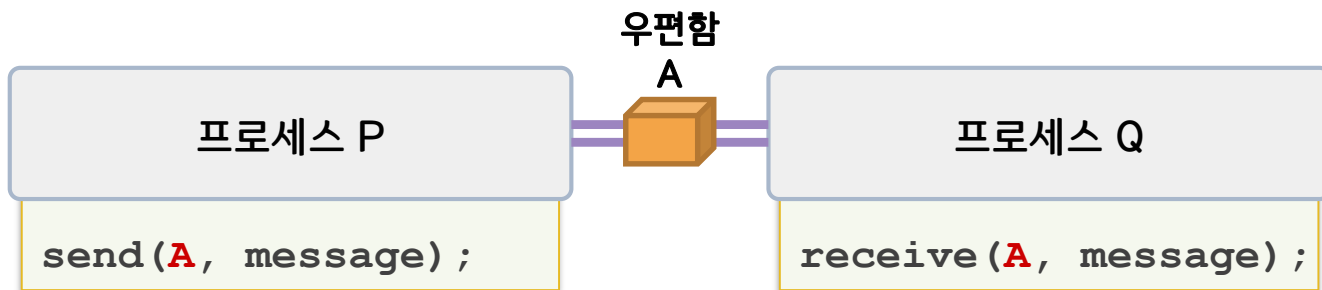
• 비대칭형



메시지 시스템 기법

■ 간접 통신(우편함)

- 메시지 전달 연산에 우편함 이름을 명시

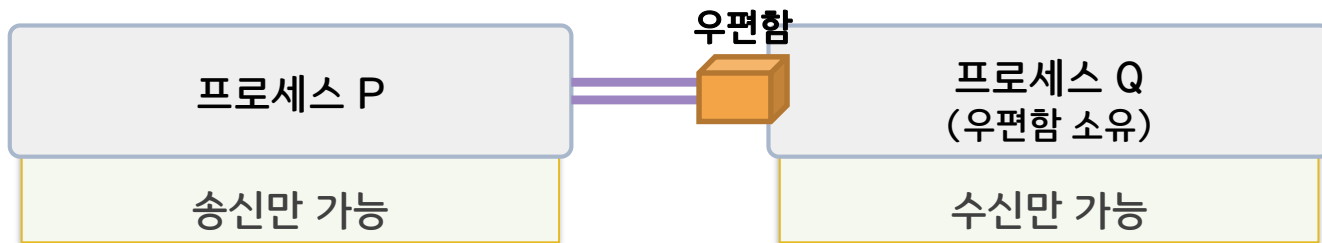


- 통신 링크는 공유 우편함이 있는 경우에만 설정됨
- 하나의 링크는 2개 이상의 프로세스들과 연관될 수 있으며 각 통신 프로세스 쌍 사이에는 여러 링크가 존재 가능
- 링크는 단방향 또는 양방향임

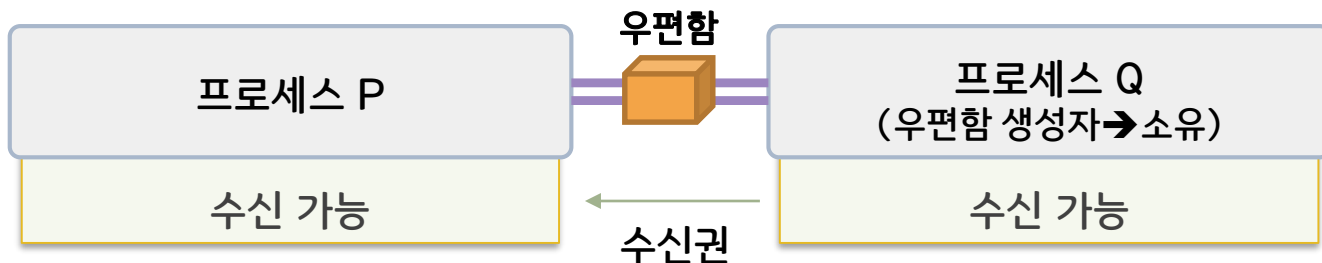
메시지 시스템 기법

■ 간접 통신(우편함)

- 우편함이 프로세스에 소속되는 경우



- 우편함이 운영체제에 소속되는 경우



메시지 시스템 기법

■ 링크의 용량

• '0' 용량



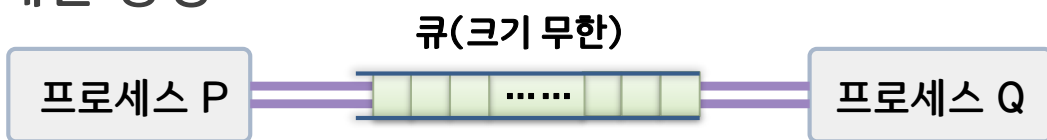
» 동기화 필요

• 제한된 용량



» 큐가 가득 차면
송신자는 대기

• 무제한 용량



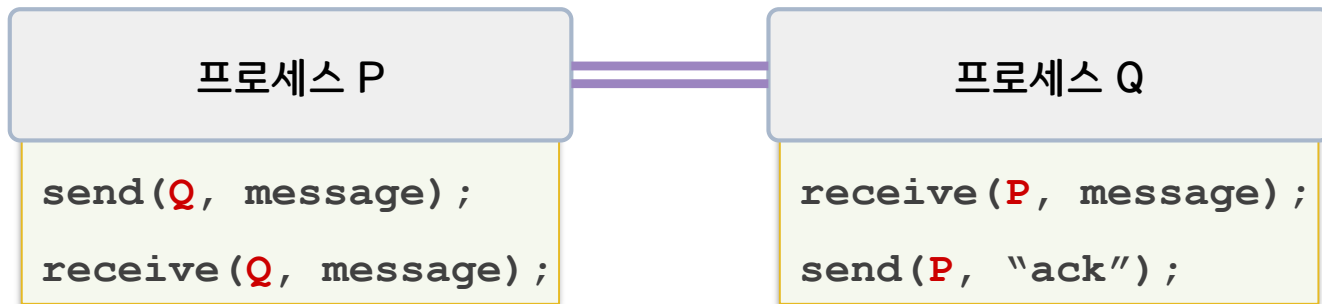
» 송신자는 대기할
필요가 없음

자동
버퍼링

메시지 시스템 기법

■ 링크의 용량

- '0' 용량이 아닌 경우 메시지 도착 여부의 인지 방법



» 비동기적 통신

ㅇ 메시지 시스템 기법

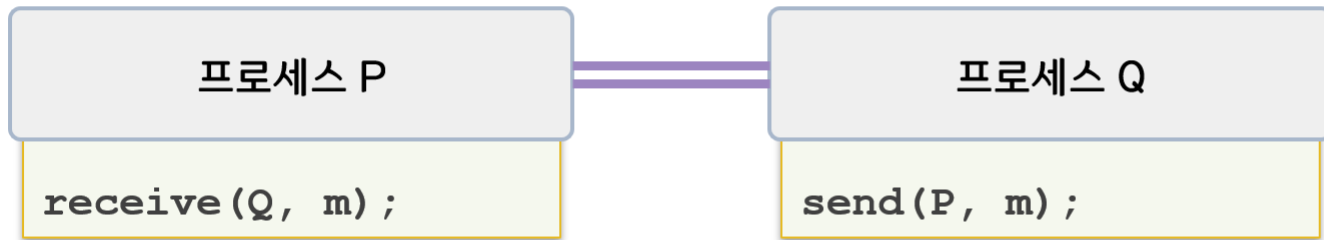
■ 예외조건 처리

- 프로세스가 종료된 경우
- 메시지를 상실한 경우
- 메시지가 혼합된 경우

메시지 시스템 기법

■ 예외조건 처리 – 프로세스 종료

- 송신 프로세스 Q가 종료된 경우

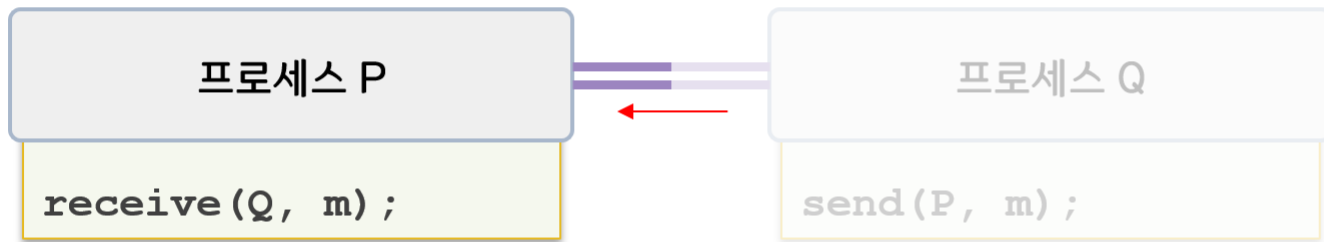


- 수신 프로세스 Q가 종료된 경우

메시지 시스템 기법

■ 예외조건 처리 – 프로세스 종료

- 송신 프로세스 Q가 종료된 경우



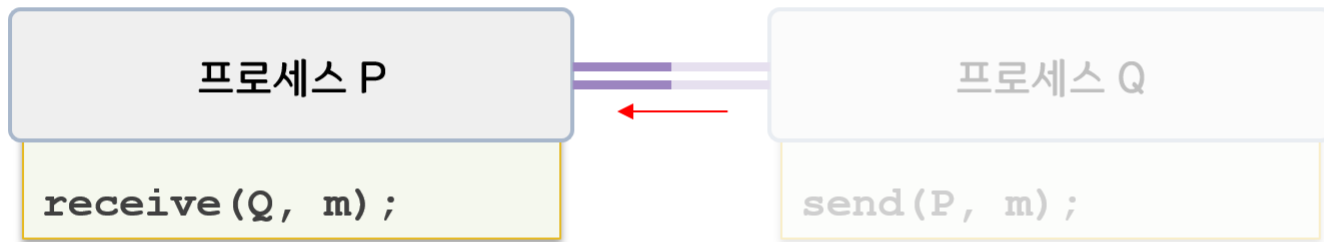
- » P를 종료
- » Q가 종료한 사실을 P에 알림

- 수신 프로세스 Q가 종료된 경우

메시지 시스템 기법

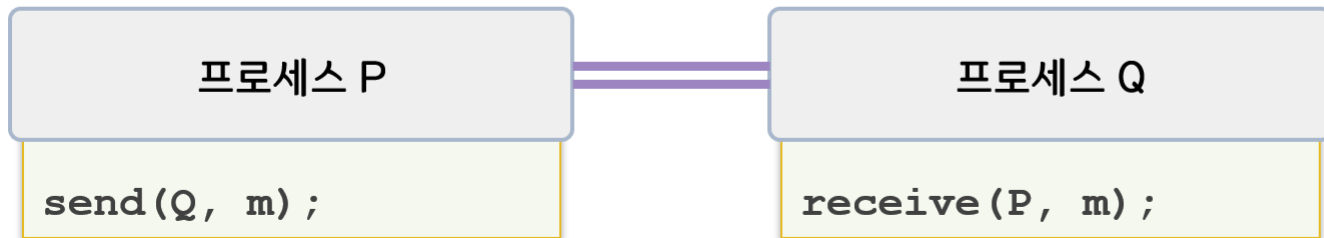
■ 예외조건 처리 – 프로세스 종료

- 송신 프로세스 Q가 종료된 경우



- » P를 종료
- » Q가 종료한 사실을 P에 알림

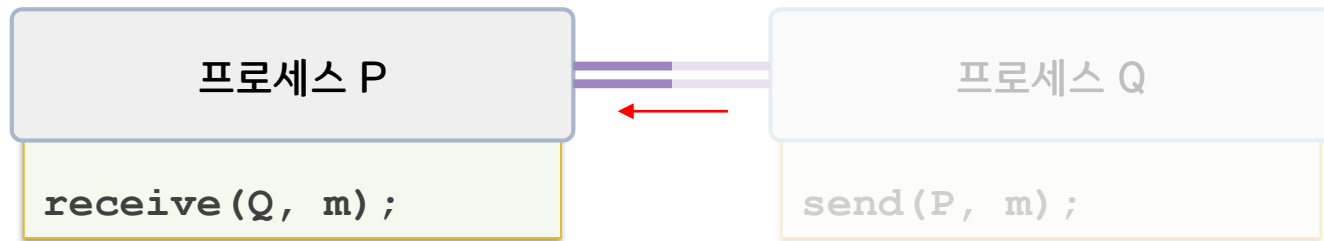
- 수신 프로세스 Q가 종료된 경우



메시지 시스템 기법

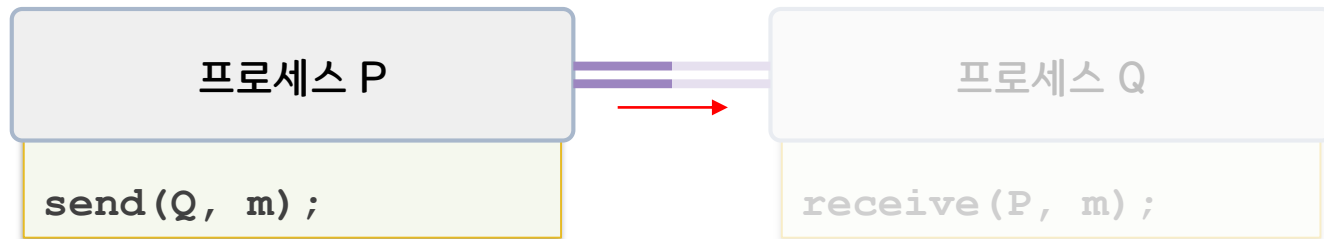
■ 예외조건 처리 – 프로세스 종료

- 송신 프로세스 Q가 종료된 경우



- » P를 종료
- » Q가 종료한 사실을 P에 알림

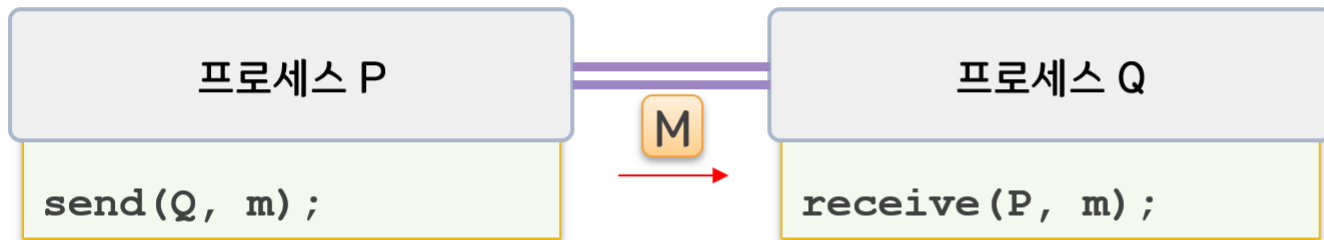
- 수신 프로세스 Q가 종료된 경우



- ↪ 버퍼가 없는 경우
- » P를 종료
- » Q가 종료한 사실을 P에 알림

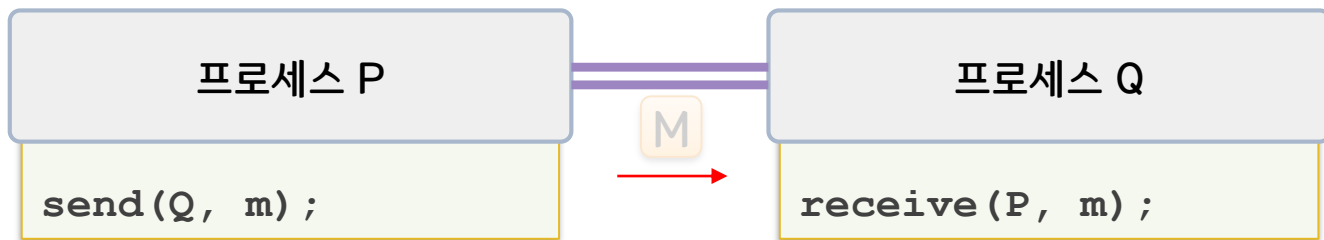
메시지 시스템 기법

예외조건 처리 – 메시지 상실



메시지 시스템 기법

예외조건 처리 – 메시지 상실



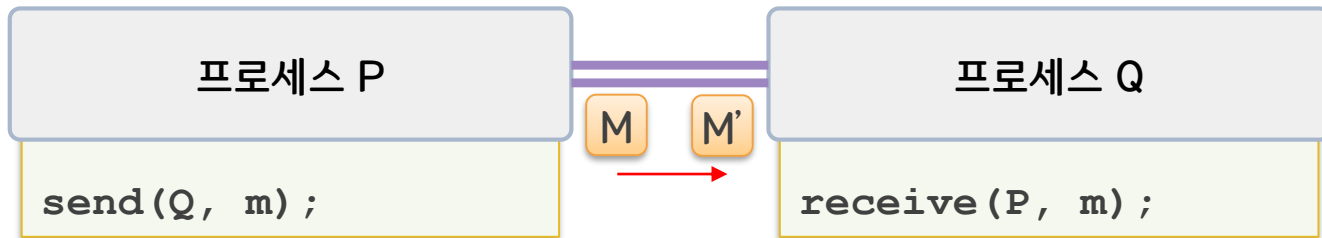
- 운영체제가 탐지 후 메시지 재전송
- 송신 프로세스가 탐지 후 메시지 재전송
- 운영체제가 탐지 후 송신 프로세스에게 통지

★ 탐지방법

시간제한

메시지 시스템 기법

예외조건 처리 – 메시지 혼합



- 오류 탐지 후 재전송

★ 탐지방법

checksum



강의를 마쳤습니다.

다음시간에는
6강. 교착상태 I