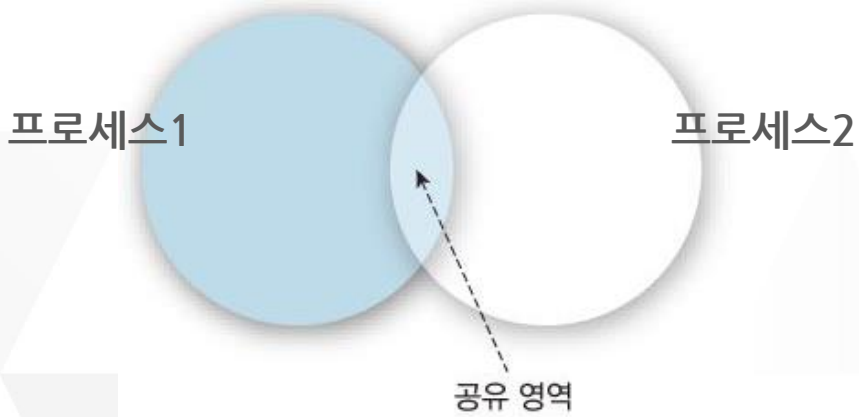
운영체제 4주차 1차시

1 병행 프로세스의 의미

1 병행 프로세스의 의미

1 개념

운영체제가 프로세서를 빠르게 전환,
프로세서 시간 나눠 마치 프로세스 여러 개를
동시에 실행하는 것처럼 보이게 하는 것



1 병행 프로세스의 의미

2 종류

독립 프로세스

- 단일 처리 시스템에서 수행하는 병행 프로세스, 다른 프로세스에 영향 주고받지 않으면서 독립 실행
- 다른 프로세스, 데이터와 상태 공유 않고 동작도 재현 가능
- 주어진 초기값에 따라 항상 동일한 결과
- 중지 후 변동 없이 다시 시작 가능
- 독립 실행할 수 있는 프로세스

1 병행 프로세스의 의미

2 종류

협력 프로세스

- 다른 프로세스와 상호작용하며 특정 기능 수행하는 비동기적 프로세스
- 제한된 컴퓨터 자원의 효율성 증대, 계산 속도 향상, 모듈적 구성 강화, 개별 사용자의 여러 작업 동시에 수행 편의성 제공에 사용
- 프로세스 하나가 파일에서 읽기 수행 동안 다른 프로세스가 해당 파일에 쓰게 하면 서로 영향
- 병행 프로세스들이 입출력장치, 메모리, 프로세서 등 자원을 서로 사용 시 충돌 발생

1 병행 프로세스의 의미

2 종류

협력 프로세스

- 충돌을 피하기 위한 프로세스의 상호작용 형태

① 프로세스는 서로 인식하지 못하는 경쟁 관계 유지

- 다중 프로그래밍 환경이 대표적인 예로, 운영체제가 자원 경쟁 고려하여 동일한 디스크나 프린터로 접근 조절

1 병행 프로세스의 의미

2 종류

협력 프로세스

- 충돌을 피하기 위한 프로세스의 상호작용 형태

② 프로세스는 입출력 버스를 비롯한 개체를 공유하는 단계에서 간접적으로 서로 관계 인식

- 이때 다른 프로세스에서 얻은 정보에 의존, 프로세스의 타이밍에 영향
- 프로세스들은 개체 공유에 따른 협력 필요

1 병행 프로세스의 의미

2 종류

협력 프로세스

- 충돌을 피하기 위한 프로세스의 상호작용 형태

  ③ 프로세스에는 서로 인식하고 프로세스끼리 통신할 수 있는 기본 함수 있음
  - 프로세스가 서로 협력 관계에 있으면 직접 통신 가능
  - 병행해서 함께 동작 가능

3 병행 프로세스의 해결 과제

병행성

- 여러 프로세스를 이용하여 작업을 수행하는 것
- 시스템 신뢰도 높이고 처리 속도 개선,
  처리 능력 높이는 데 중요

3 병행 프로세스의 해결 과제

병행 프로세스의 문제

- 공유 자원 상호 배타적 사용
  (프린터, 통신망 등은 한순간에 프로세스 하나만 사용)
- 병행 프로세스 간의 협력이나 동기화
  (상호배제도 동기화의 한 형태)
- 두 프로세스 간 데이터 교환을 위한 통신
- 동시에 수행하는 다른 프로세스의 실행 속도와
  관계 없이 항상 일정한 실행 결과 보장

3 병행 프로세스의 해결 과제

병행 프로세스의 문제

- 교착 상태 해결, 병행 프로세스들의 병렬 처리 능력 극대화 실행 검증 문제 해결
- 병행 프로세스 수행 과정에서 발생하는 상호배제 보장

2 선행 그래프

2 선행 그래프

1 의미

- 선행 제약의 논리적 표현
- 프로세스
  - 프로세스 집합과 이것의 선행 제약 두 가지 요소로 정의

※ 선행 제약이란?

- 프로세스를 순서대로 다른 상태로 옮기는 것 (프로세스에 선행 제약이 없으면 이 둘은 독립적이므로 병행 실행 가능)

2 선행 그래프

1 의미

- 순차적 활동을 표현하는 방향성 비순환 그래프
- 선행 그래프에서 노드는 소프트웨어 작업이거나 동시에 실행할 수 있는 프로그램 명령

2 선행 그래프

1 의미

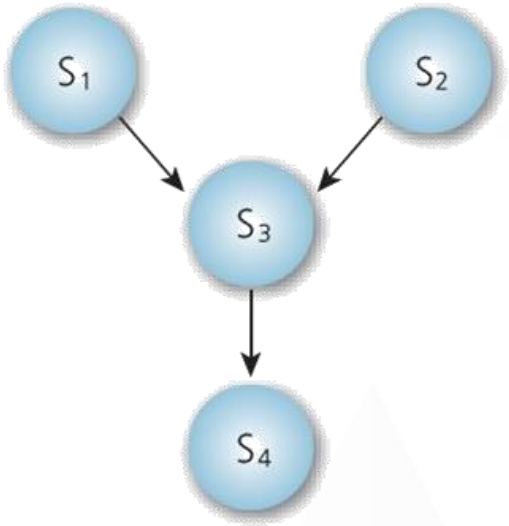
a := x + y; → S_1

b := z + 1; → S_2

c := a − b; → S_3

w := c + 1; → S_4

(a) 알고리즘

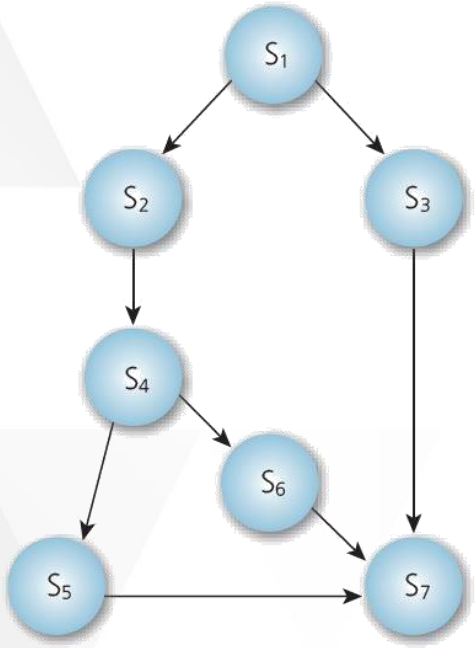


(b) 선행 그래프

※ 출처 : 그림으로 배우는 구조와 원리 운영체제, 구현회, 한빛아카데미

2 선행 그래프

1 의미



(a) 선행 그래프

- S_2와 S_3은 S_1이 끝난 후에 수행한다.
- S_4는 S_2가 끝난 후에 수행한다.
- S_5와 S_6은 S_4가 끝난 후에 수행한다.
- S_7은 S_5, S_6, S_3이 끝난 후에 수행하고, S_3은 S_2, S_4, S_5, S_6과 병행하여 수행할 수 있다.

(b) 선행 관계

※ 출처 : 그림으로 배우는 구조와 원리 운영체제, 구현회, 한빛아카데미

3 언어적 표현과 병행문장

3 언어적 표현과 병행문장

1 Fork 와 Join 구조

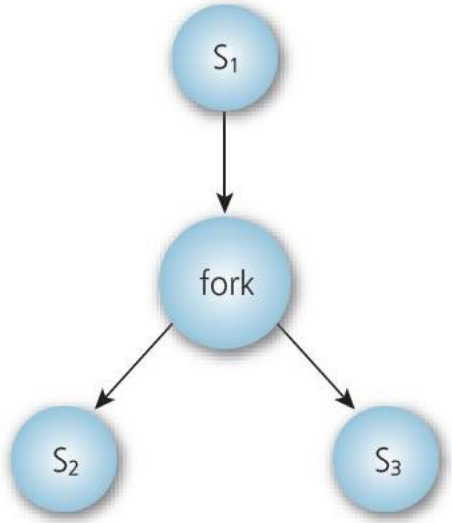
- 선행 그래프는 연산의 선행 제약 정의에 유용하지만, 2차원이라 프로그램에는 사용 곤란
- 선행 관계 명시 위해 Fork와 Join 구조, 병행 문장 (Parbegin/Parend) 등 다른 방법 필요
- 콘웨이 (1963년)와 데니스Dennis(1966년), 혼 (1966년)이 소개
- Fork와 Join 두 명령어 사용 최초로 병행을 언어적으로 표현

3 언어적 표현과 병행문장

2 Fork 명령어

```
        S1;
        fork L;
        S2;
        .
        .
        .
L :     S3;
```

(a) 알고리즘



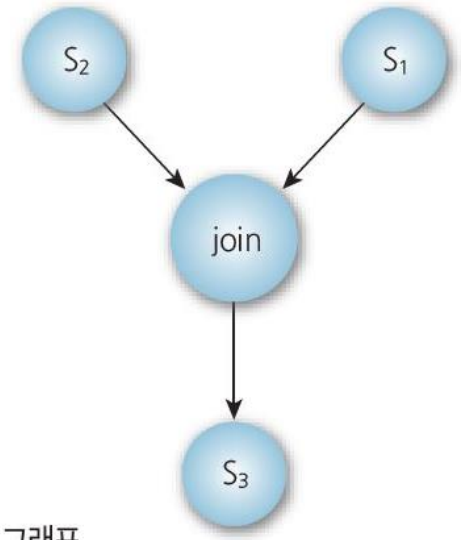
(b) 선행 그래프

※ 출처 : 그림으로 배우는 구조와 원리 운영체제, 구현회, 한빛아카데미

3 Join 명령어

```
        count := 2;
        fork L1;
        :
        S1;
        goto L2;
L1 :    S2;
L2 :    join count;
        S3;
```

(a) 알고리즘



(b) 선행 그래프

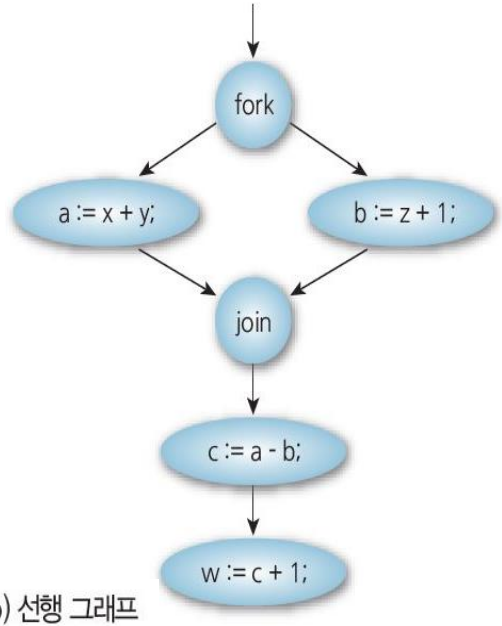
※ 출처 : 그림으로 배우는 구조와 원리 운영체제, 구현회, 한빛아카데미

3 언어적 표현과 병행문장

4 Fork와 join 구조의 알고리즘과 선행그래프

```
        count := 2;
        fork L1;
        a := x + y;
        goto L2;
        b := z + 1;
L1 :    join count;
L2 :    c := a - b;
        w := c + 1;
```

(a) 알고리즘

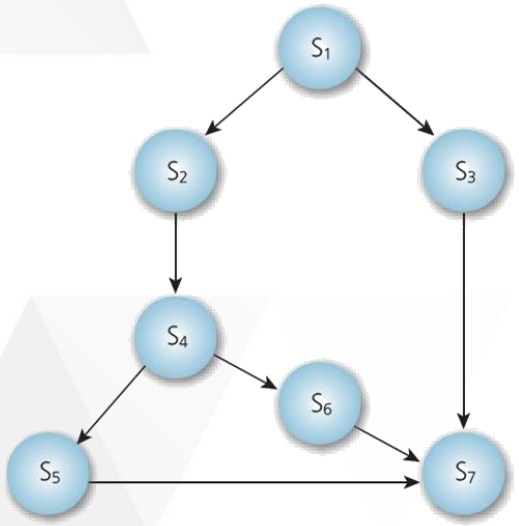


(b) 선행 그래프

※ 출처 : 그림으로 배우는 구조와 원리 운영체제, 구현회, 한빛아카데미

3 언어적 표현과 병행문장

4 Fork와 join 구조의 알고리즘과 선행그래프



(a) 선행 그래프

```
        S1;
        count := 3;
        fork L1;
        S2;
        S4;
        fork L2;
        S5;
        goto L3;
L2  :   S6;
L1  :   goto L3;
        S3;
L3  :   join count;
        S7;
```

(b) 알고리즘

※ 출처 : 그림으로 배우는 구조와 원리 운영체제, 구현회, 한빛아카데미

3 언어적 표현과 병행문장

5 병행 문장

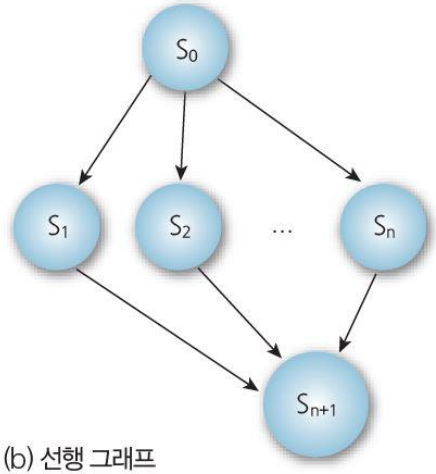
- 하나의 프로세스가 여러 병렬 프로세스로 퍼졌다가 다시 하나로 뭉쳐지는 것을 나타냄
- 대표적인 예 :
  다익스트라(1965년)가 제안한 Parbegin/Parend

5 병행 문장

일반적인 형태

S_0; parbegin S_1; S_2; . . . . . ; S_n; parend; S_{n+1};

(a) 일반 구조의 병행 문장



(b) 선행 그래프

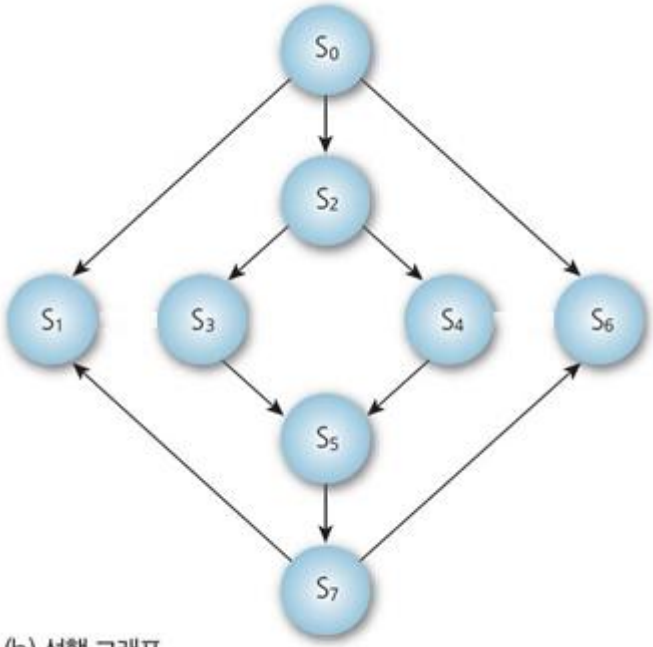
※ 출처 : 그림으로 배우는 구조와 원리 운영체제, 구현회, 한빛아카데미

3 언어적 표현과 병행문장

5 병행 문장

```
S0;
PARBEGIN
   S1;
   BEGIN
      S2;
      PARBEGIN
         S3;
         S4;
      PAREND;
      S5;
   END;
   S6;
PAREND;
S7;
```

(a) 알고리즘



(b) 선행 그래프

※ 출처 : 그림으로 배우는 구조와 원리 운영체제, 구현회, 한빛아카데미

3 언어적 표현과 병행문장

5 병행 문장

```
a := x + y;
b := z + 1;
c := a - b;
w := c + 1;
```

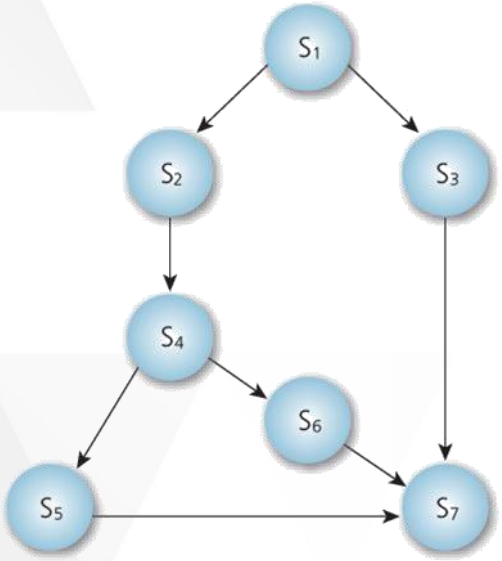
(a) 간단한 산술 알고리즘

```
parbegin
    a := x + y;
    b := z + 1;
parend;
    c := a - b;
    w := c + 1;
```

(b) parbegin/parend 구조 알고리즘

※ 출처 : 그림으로 배우는 구조와 원리 운영체제, 구현회, 한빛아카데미

5 병행 문장



(a) 선행 그래프

※ 출처 : 그림으로 배우는 구조와 원리 운영체제, 구현회, 한빛아카데미

```
S1;
parbegin
   S3;
   begin
      S2;
      S4;
      parbegin
      S5;
      S6;
      parend;
   end;
parend;
S7;
```

(b) 알고리즘