

배치 작업 자동화 스케줄러

CIB 플랫폼부 / 인턴 김시환

Contents

- I. 배치 작업 자동화 스케줄러
- II. 진행 경과
- III. 개발 환경
- IV. 기능 설명
- V. 시스템 구성도
- VI. 처리 흐름도

I. 배치 작업 자동화 스케줄러



배치 작업 자동화 스케줄러

배치 작업 (Batch job) : 실시간으로 처리하는 것이 아니라 일괄 처리하는 작업

스케줄러 : 컴퓨터 시스템에 입력된 작업들이 수행될 수 있도록 작업에 자원을 적절히 할당하고, 작업이 실행될 수 있도록 관리하는 프로그램

활용 예시 : 매일 퇴근 전에 작성했던 파일을 취합해서 저장

II. 진행 경과



진행 경과

분류	내용	예상 일정	실제 완료 일시
서버 개발	작업 등록/실행여부 판단 구현 - 초기, 웹 브라우저 상에서 동작하도록 구현 - Javascript로 구현	8.4 (금)	8.4 (금)
	작업 실행 구현 - React, Express로 이전 - child_process를 활용하여 스크립트 실행	8.9 (수)	8.10 (목)
	작업 수정/삭제 구현 - DB와 연동한 기본적인 등록/수정/삭제 구현	8.14(월)	8.14 (월)
	스케줄링 알고리즘 구현 - 스케줄링 알고리즘 자료 조사 후 선정 - Heap 자료구조 및 메서드 구현 - Blocking / Non-Blocking 방식 구현	8.23 (수)	8.22 (화)
화면 개발	작업 테이블 구현 실행 흐름도 구현 실행 로그 화면 구현	8.25 (금)	8.28 (월)

III. 개발 환경



개발 환경

언어 : JavaScript (ES6+)

프론트엔드 개발 (화면) : React.js (v18.2.0)

백엔드 개발 (서버) : Express.js (v4.18.2)

DB : MySQL

IV. 기능 설명

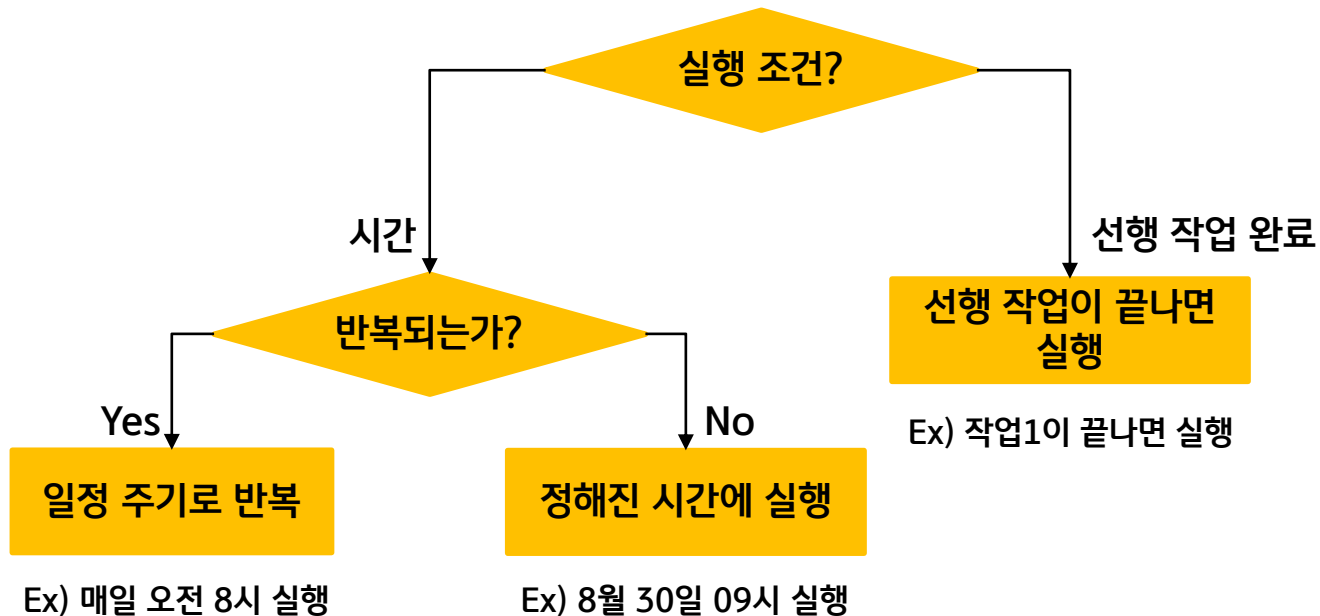


기능 개요

- 핵심 기능 (~8.22)
 - 작업 실행 가능 여부 판단
 - 작업 실행
- 부가 기능 (~8.28)
 - 작업 등록/수정/삭제
 - 실행 흐름
 - 실행 로그

핵심 기능 - 작업 실행 가능 여부 판단 (1)

- 작업 실행 가능 여부를 판단하기 위해 먼저 작업을 분류해야 한다.
- 분류 조건
 1. 실행 조건이 무엇인가?
 2. 실행 조건이 시간일 경우, 반복되는가?
- 위 조건에 따라 작업을 분류하면 3가지로 나눌 수 있다.

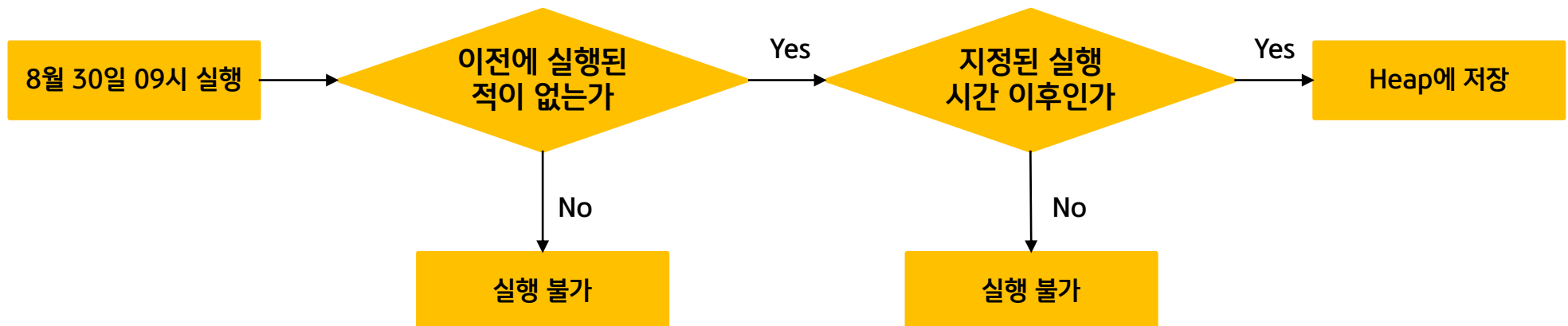


핵심 기능 - 작업 실행 가능 여부 판단 (1)



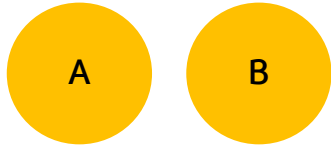
핵심 기능 - 작업 실행 가능 여부 판단 (2)

- 분류된 작업마다 정해진 조건에 따라 실행 여부를 판단한다.
- 조건을 모두 만족해야 실행 가능하다고 판단한다.
- 실행 가능하다고 판단되면 Heap*에 저장한다.



핵심 기능 - 작업 실행 가능 여부 판단 (2)

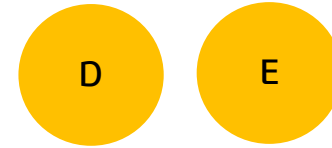
일정 주기로 반복



정해진 시간에 실행

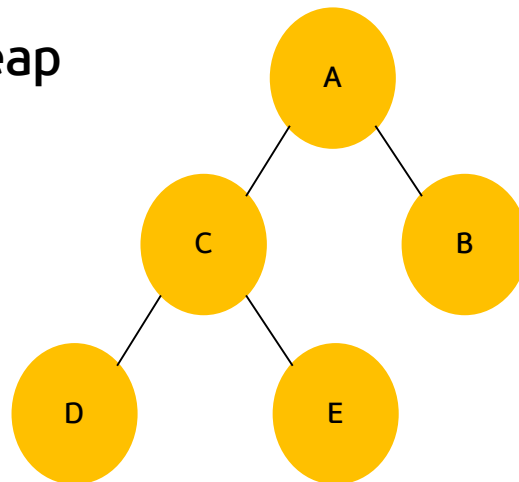


선행 작업이 끝나면
실행



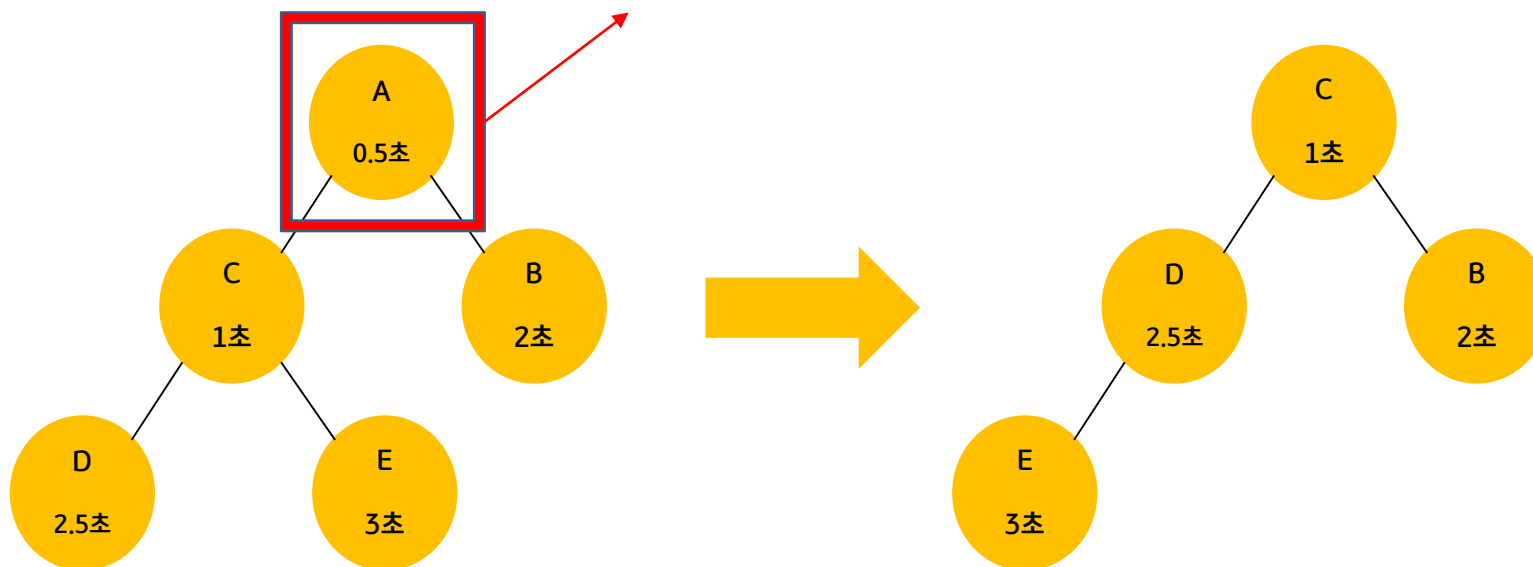
실행 가능 여부 판단

Heap



Heap

- Heap
 - 최댓값 또는 최솟값을 찾아내는 연산을 빠르게 하기 위해 고안된 이진트리
- Heap을 사용한 이유
 - Shortest-Job-First 스케줄링 알고리즘 사용하기 위해
 - Shortest-Job-First : 실행 완료까지 가장 짧게 걸리는 작업을 우선으로 실행
 - 실행 시간이 가장 짧은 작업이 항상 최상단에 위치한다.
 - 따라서 실행 시간이 가장 짧은 작업을 빠르게 찾을 수 있다.



핵심 기능 - 작업 실행 (1)

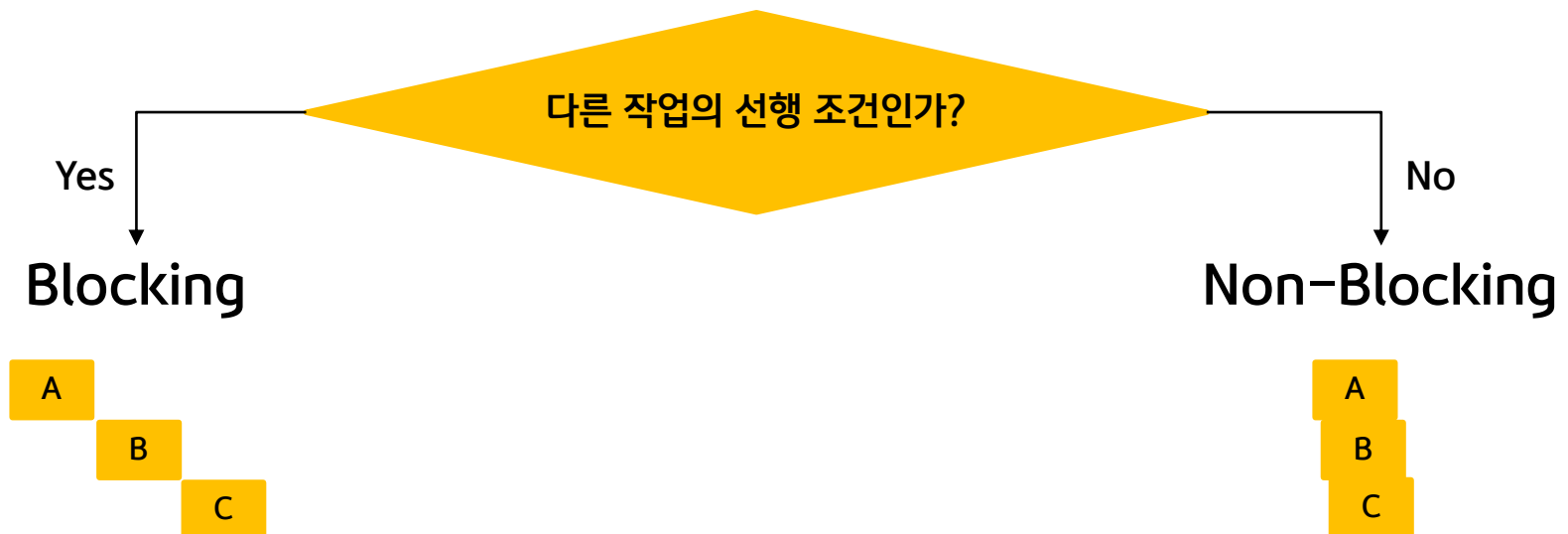
- Heap에 저장된 작업들을 실행하는 방식을 결정해야 한다.

1. Blocking

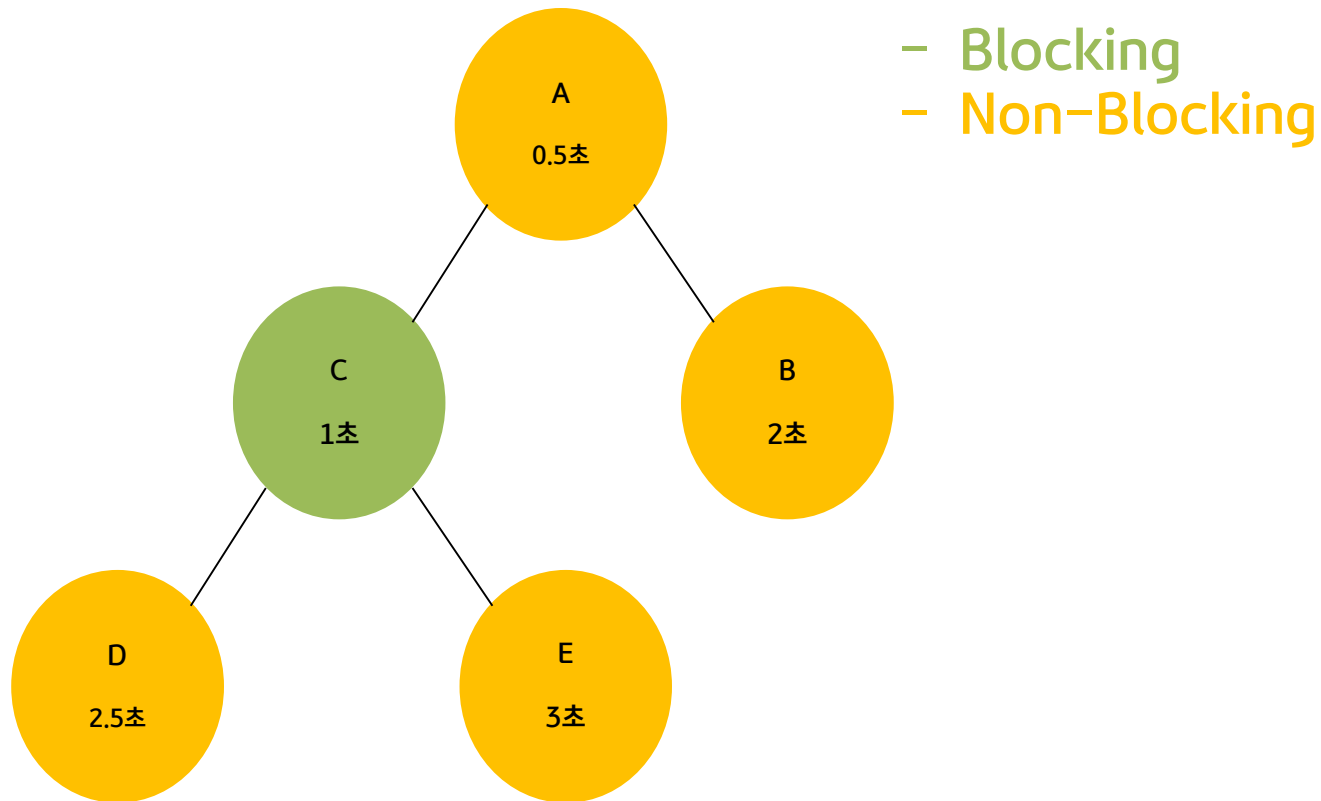
- 한 작업이 실행되기 시작하면 완료될 때까지 다른 작업이 실행되지 않음
- 다른 작업의 선행 조건인 작업을 실행할 때 사용
- 선행 조건이 완료되고 난 후, 다음 작업이 실행되어야 하기 때문

2. Non-Blocking

- 한 작업이 실행되더라도 다른 작업 실행 가능
- 다른 작업의 선행조건이 아닌 작업을 실행할 때 사용

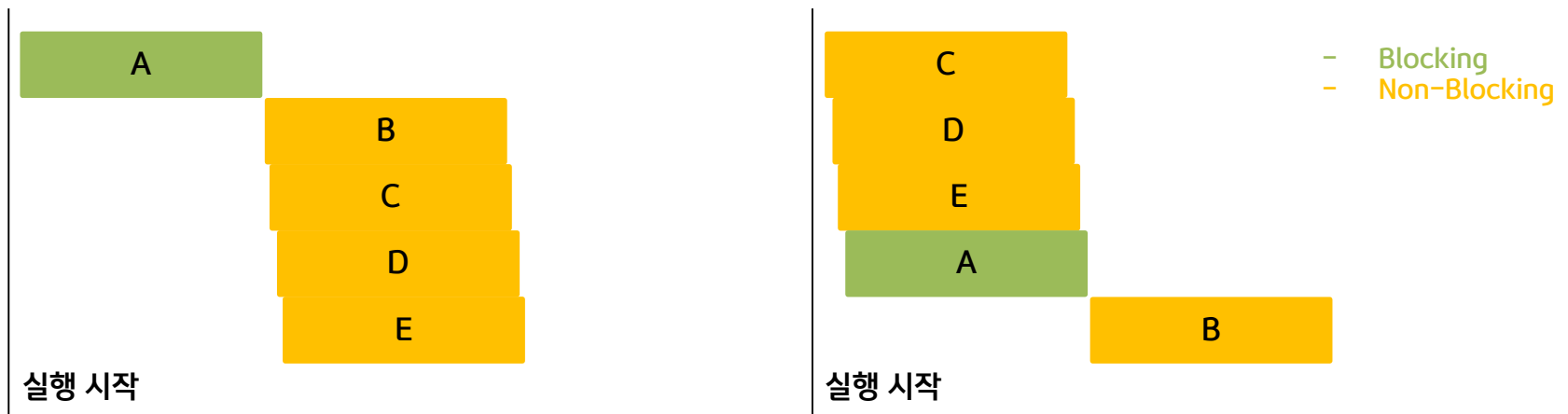


핵심 기능 - 작업 실행 (1)



핵심 기능 - 작업 실행 (2)

- 문제점 : Blocking 방식으로 실행되는 작업의 실행시간이 짧아서 가장 먼저 실행된다면, 이 작업이 끝나기를 기다릴 필요가 없는 작업들도 대기하는 상황이 발생한다.
- 아래 그림에서 모든 작업의 실행 시간이 5초이고, 작업 B의 선행 조건이 A라고 가정

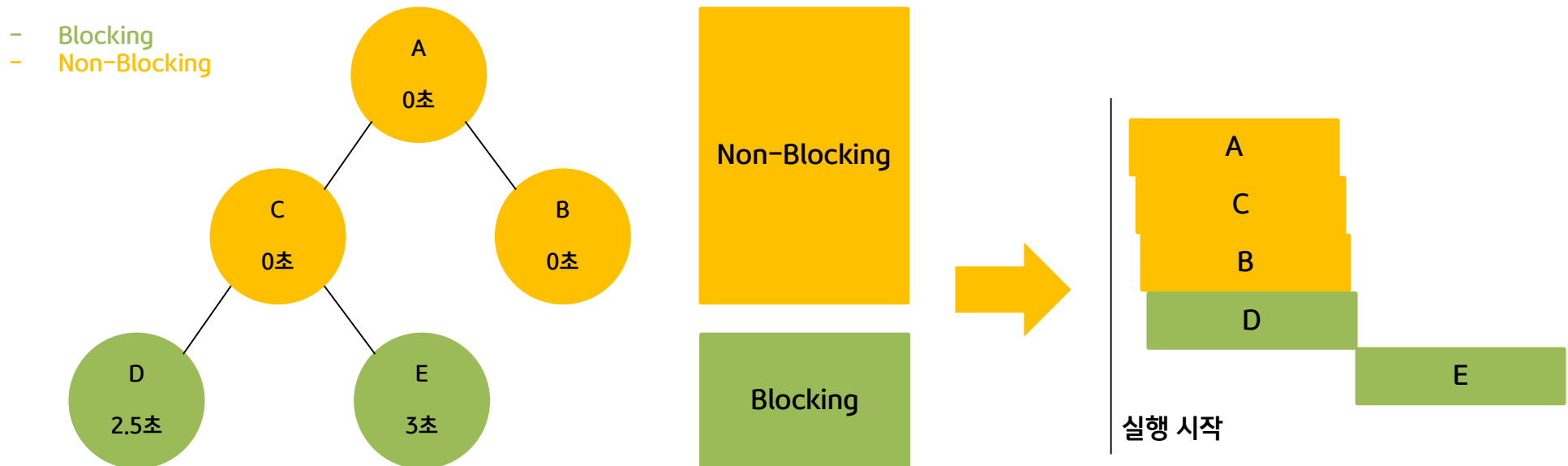


평균 대기 시간	$(0+5+5+5+5) / 5$ ≈ 4 sec	$(0+0+0+0+5) / 5$ ≈ 1sec
평균 완료 시간	$(5+10+10+10+10) / 5$ ≈ 9sec	$(5+5+5+5+10) / 5$ ≈ 6sec

결론 : Heap에 저장된 작업 중 다른 작업의 선행 조건인 작업(Blocking)은 최대한 늦게 실행

핵심 기능 - 작업 실행 (3)

- 방법 : Non-Blocking 작업들의 실행시간을 0으로 설정한다.
- 이유 : 실제 실행 시간과 설정된 실행시간이 다르더라도 Non-Blocking 방식으로 실행되는 작업은 실행되고 바로 다음 작업이 실행되기 때문에 이후 실행에 영향을 미치지 않음
- 이를 통해, Non-Blocking 작업들은 Heap 상단부에 위치하게 된다.
- Non-Blocking 작업이 모두 실행되기 전까지는 Blocking 작업이 실행되지 않는다.



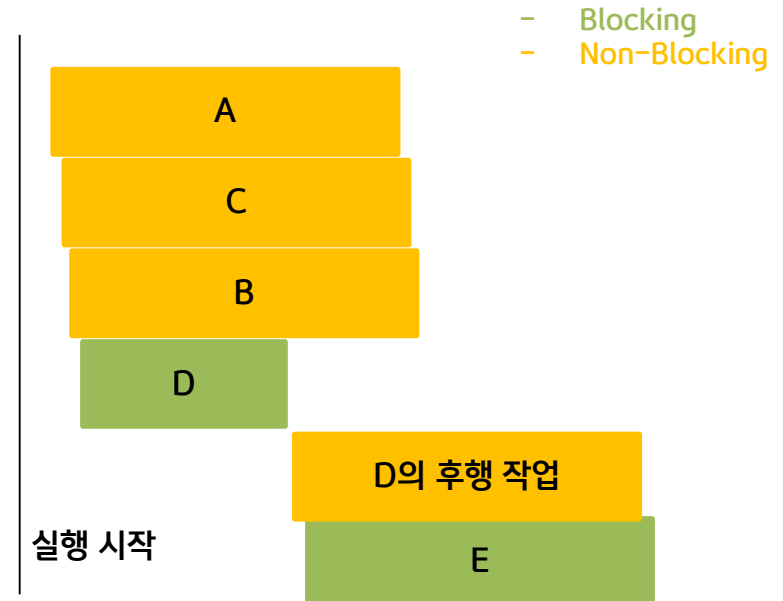
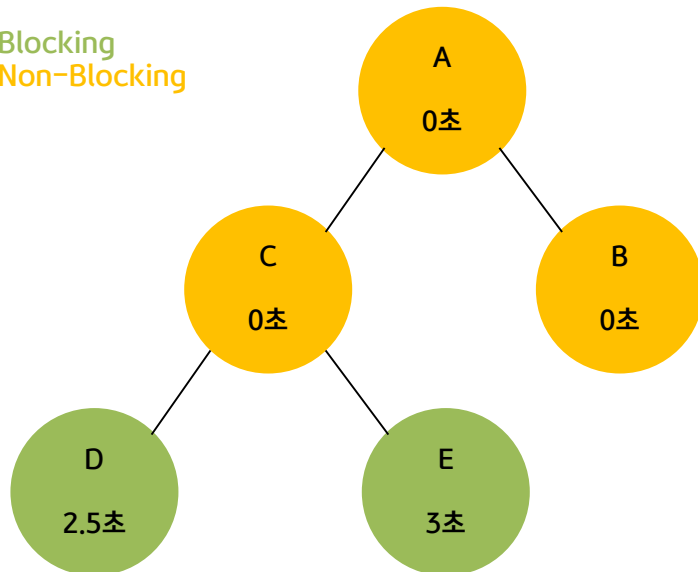
핵심 기능 정리 (1)

- 등록된 작업들 중 실행 가능하다고 판단되는 작업은 Heap에 저장
- 이때, Non-Blocking 방식으로 실행될 작업들은 실행 시간이 0으로 설정되어 Heap 상단에 위치
- Heap 최상단에 있는 작업을 꺼내면서 작업 별 정해진 방식에 따라 실행
- 위 규칙만 지키면 자연스럽게 자원 사용을 최소화하는 방법으로 작업을 실행 할 수 있다.

핵심 기능 정리 (2)

- Heap에 존재하는 작업들 중 실행되는 순서는 다음과 같다.
- Heap에 Non-Blocking 작업이 없을 때까지 Non-Blocking 작업을 먼저 실행하고,
- Non-Blocking 작업을 모두 실행했다면, Blocking 작업 실행
- Blocking 작업을 실행할 때에는 실행 시간이 짧은 순으로 실행 (Shortest-Job First)

- Blocking
- Non-Blocking



부가 기능 - 작업 등록/수정/삭제

- 작업 관리 화면 제공
- 등록/수정/삭제 기능 구현

작업 등록

이름

A

주기

조건

8

월

29

일

8

시

10

분

파일 선택

b.sh

제출

취소



실행 대기중인 작업

이름	실행 시간	선행 조건	파일명	실행 흐름	수정/삭제
A	8월 29일 8시 10분	-	b.sh	<div>보기</div>	<div>수정</div> <div>삭제</div>

부가 기능 - 실행 흐름

- 실행 흐름도를 제공
- 선택한 작업이 선행 조건이 있을 경우, 선행 조건을 보여준다.
- 이어서 실행 되는 작업이 있다면, 해당 작업을 보여준다.

실행 대기중인 작업

이름	실행 시간	선행 조건	파일명	실행 흐름	수정/삭제
A	8월 29일 8시 10분	-	b.sh	보기	수정 삭제
B	-	A	a.sh	보기	수정 삭제
C	-	B	c.sh	보기	수정 삭제



실행 흐름

A

↓

B

↓

C

부가 기능 - 실행 로그

- 실행 로그 제공
- 어떤 작업이 언제 완료되었는지 확인할 수 있다.

실행 로그

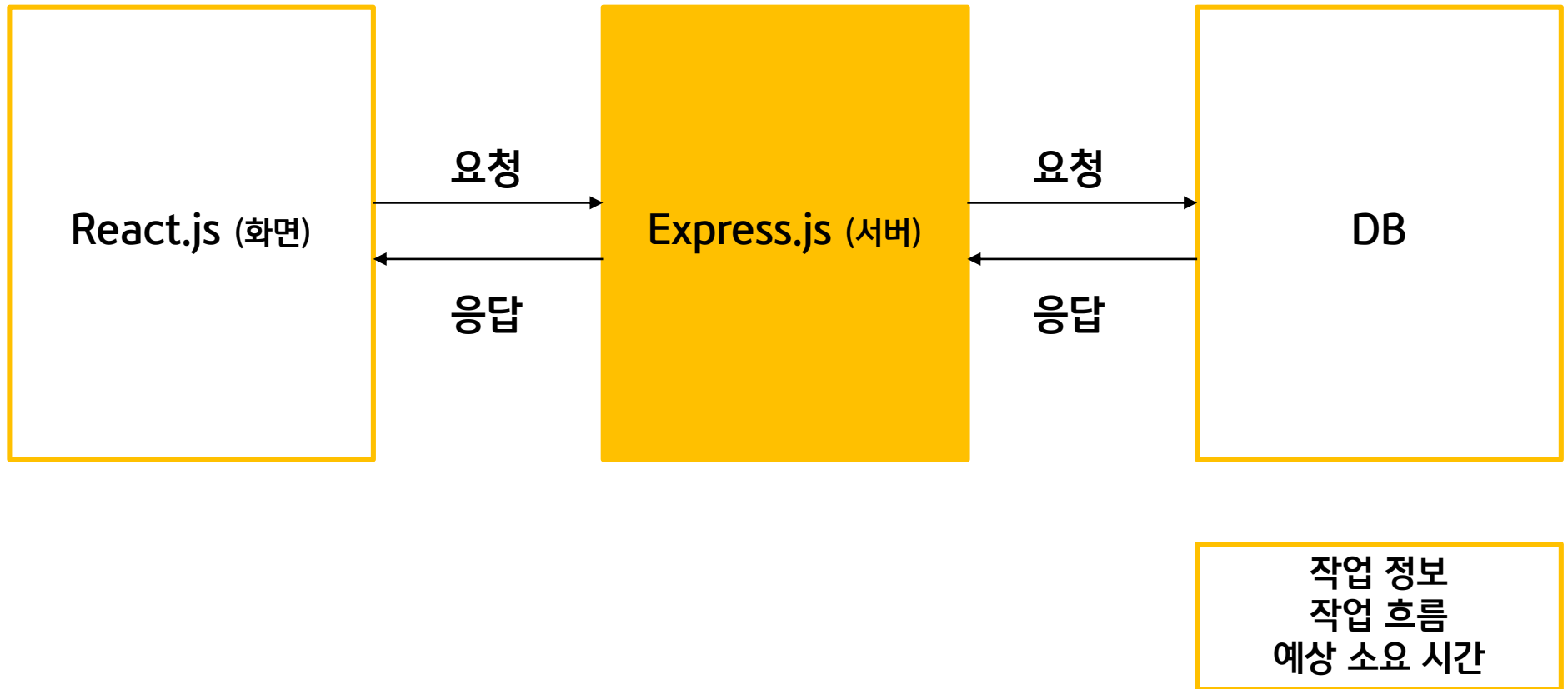
새로 고침

c에 저장 완료 - Tue Aug 29 2023 16:26:31 GMT+0900 (대한민국 표준시)
e에 저장 완료 - Tue Aug 29 2023 16:26:31 GMT+0900 (대한민국 표준시)
d에 저장 완료 - Tue Aug 29 2023 16:26:31 GMT+0900 (대한민국 표준시)
a에 저장 완료 - Tue Aug 29 2023 16:26:33 GMT+0900 (대한민국 표준시)
f에 저장 완료 - Tue Aug 29 2023 16:26:36 GMT+0900 (대한민국 표준시)

V. 시스템 구성도



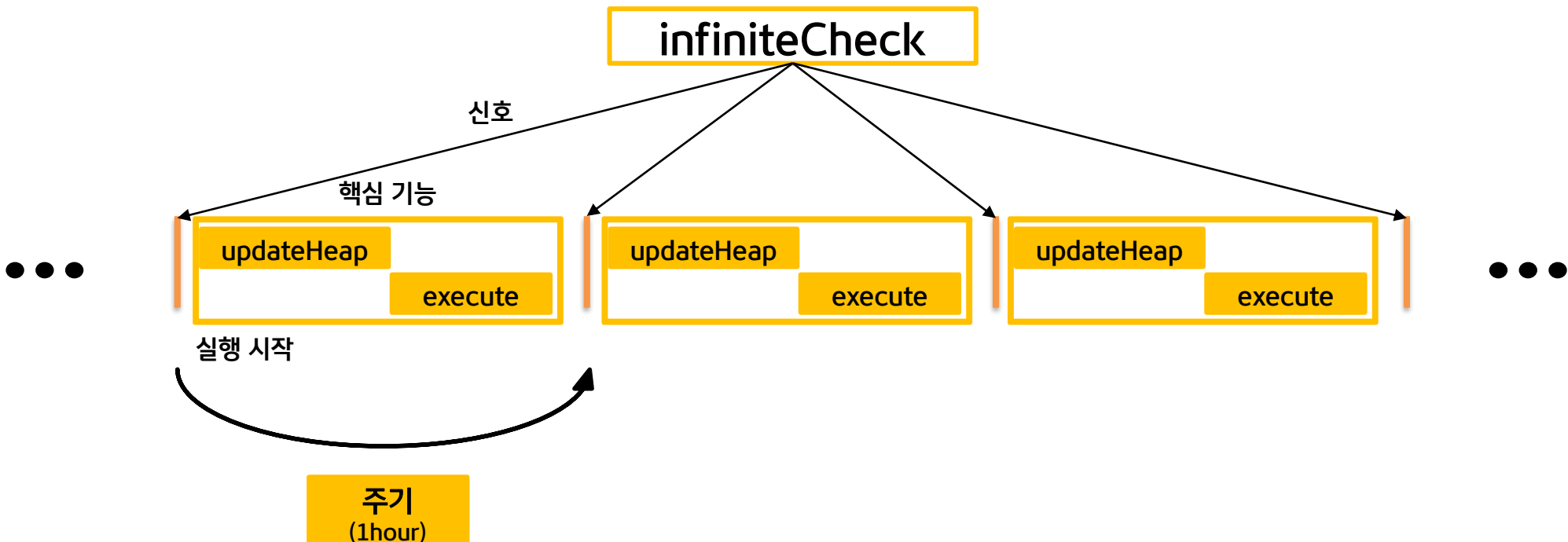
시스템 구성도 (1)



시스템 구성도 (2)

infiniteCheck

- 정해진 주기마다 신호를 주어 핵심 기능의 시작 시기를 알려준다.
- 사용자가 종료하기 전까지는 무한반복된다.



시스템 구성도 (3)

핵심 기능



updateHeap

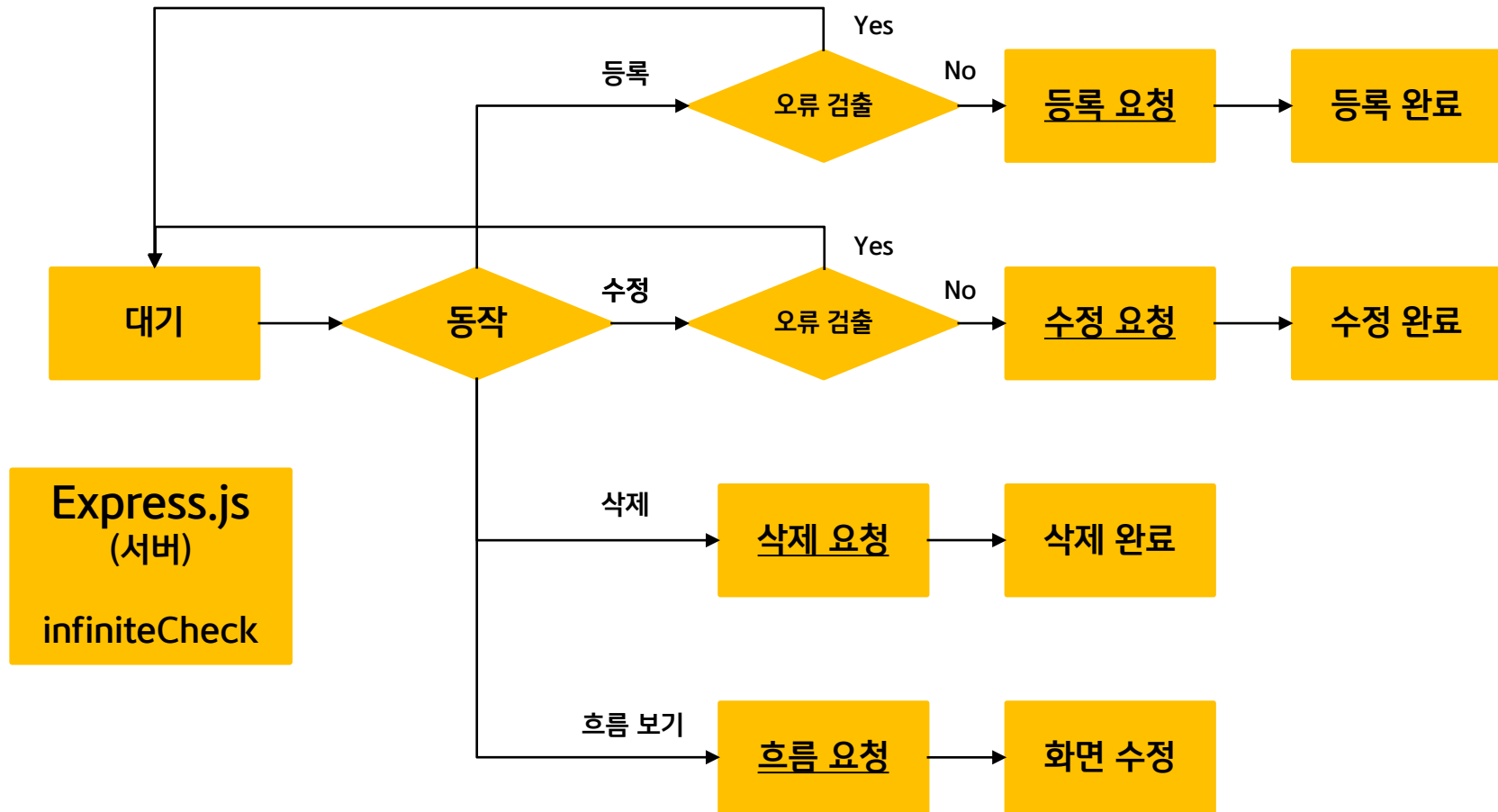
- 작업의 실행 가능 여부 판단
- 실행 가능하다고 판단되는 작업을 Heap에 저장

execute

- 작업에 부여된 실행방식 (Blocking / Non-Blocking) 으로 작업 실행

VI. 처리 흐름도

처리 흐름도



감사합니다