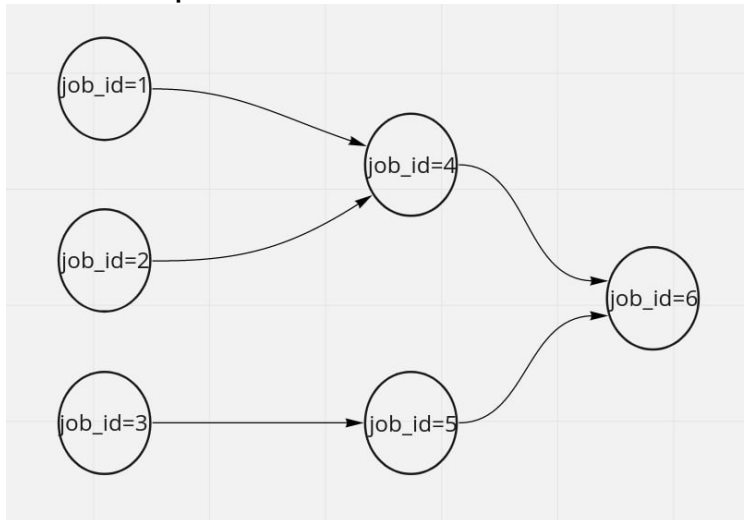


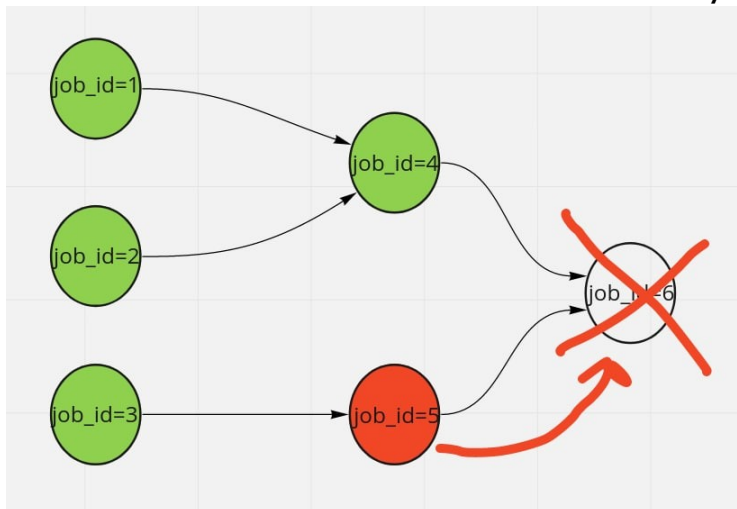
Создание планировщика DAG*’а «джобов» (jobs)**

На языке C\C++ написать программу, которая:

1. По конфигурационному файлу в формате yaml, json или ini принимает спроектированный DAG джобов и проверяет на корректность: отсутствие циклов, наличие только одной компоненты связности, наличие стартовых и завершающих джоб. Структура описания джоб и их связей произвольная.

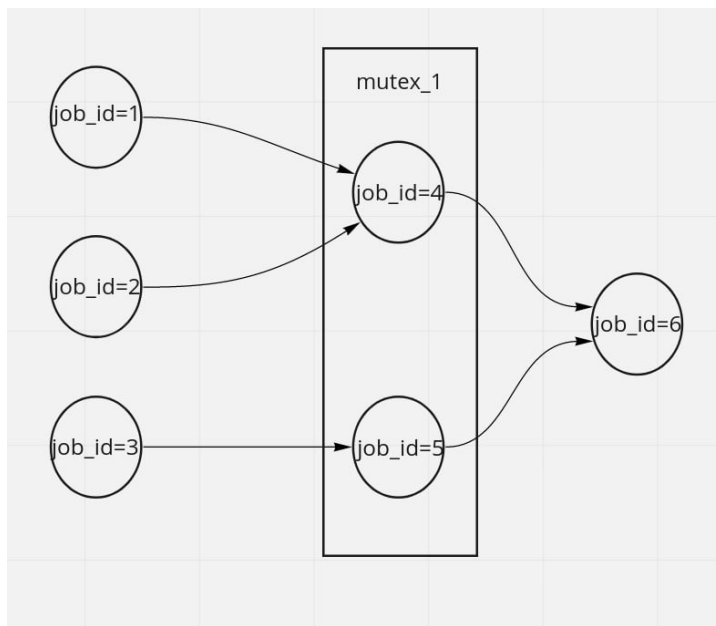


2. При завершении джобы с ошибкой, необходимо прервать выполнение всего DAG’а и всех запущенных джоб.



3. (на оценку 4) Джобы должны запускаться максимально параллельно. Должны быть ограничены параметром – максимальным числом одновременно выполняемых джоб.
4. (на оценку 5) Реализовать для джобов один из примитивов синхронизации мьютекс\семафор\барьер. То есть в конфиге дать возможность определять имена

семафоров (с их степенями)\мьютексов\барьеров и указывать их в определение джобов в конфиге. Джобы указанные с одним мьютексом могут выполняться только последовательно (в любом порядке допустимом в DAG). Джобы указанные с одним семафором могут выполняться параллельно с максимальным числом параллельно выполняемых джоб равным степени семафору. Джобы указанные с одним барьером имеют следующие свойство – зависимость от них джобы начнут выполняться не раньше того момента времени, когда выполнятся все джобы с указанным барьером.



* DAG - Directed acyclic graph. Направленный ациклический граф.

** Джоб(Job) – процесс, который зависит от результата выполнения других процессов (если он не стартовый), которые исполняются до него в DAG, и который порождает данные от которых может быть зависят другие процессы, которые исполняются после него в DAG (если он не завершающий).

Варианты:

31. Yaml\Mutex
32. Yaml\Semaphore
33. Yaml\Barrier

- 34. Json\Mutex
- 35. Json\Semaphore
- 36. Json\Barrier
- 37. Ini\Mutex
- 38. Ini\Semaphore
- 39. Ini\Barrier