Hazar pointer 정리

Lock-free에 ABA문제 발생

: 메모리 할당과 해제, 재사용 관련해서 메모리의 변경을 감지 못하는 문제

C나 C++에서는 해결해야해

어떻게? Hazard pointer (: 해제하면 위험한 포인터)

* 메모리 해제를 조정해서 바로 하지 않고, 시간이 좀 지나서 해제

개념 : hazard pointer를 담을 리스트1을 두고 모든 스레드가 공유한다. 스레드마다 자신이 접근할 리스트2가 하나 있다. 리스트2는 해당 쓰레드가 메모리를 해제할 필요가 있을 때 그 메모리를 등록하는 리스트다.

알고리즘: 어떤 쓰레드가 자신이 안전하게 엑세스 하고자 하는 메모리 블록을 리스트1에 등록하고 사용한다. 사용이 끝나면 메모리 블록을 리스트1에서 제거한다. 어떤 쓰레드가 해제할 메모리 블록이 있다면 이 메모리 블록을 리스트1에 검색하고, 없다면 해제, 있다면 리스트2에 저장한다. 나중에 메모리를 해제할 기회가 생기면 자신의 리스트2에 등록되어 있는 메모리를 리스트1 검색을 걸쳐서 삭제한다.

문제점: 동기화가 필요없는 컨테이너를 구현하는데 리스트1이 동기화가 필요하다.

연구에서 해결: 쓰레드에서 사용할 때 하자드포인터를 등록하고, 사용이 끝나면 해제한다. 쓰레드는 동시에 한, 두개의 하자드 포인터를 사용한다. (따라서 하자드 포인터는 쓰레드 수의 2배크기)

결론: 하자드 리스트에서 해제하지 않은 노드는 재사용한다. 그래서 ABA가 생기지 않는다