프로그래밍 언어론

Report #8

2017.05.31

20154073

강다현

1 Java의 JVM에서 지원하는 쓰레기 수집(garbage collection)에 대해 그 방법과 알고리즘 관점에서 조사

🏵 Garbage collection

* JVM에서 쓰지 않는 메모리를 자동으로 회수
* 동작 방법  
  JVM은 3가지 메모리 영역으로 나뉘는데 YOUNG/OLD/PERM 으로 구분된다.  
  \* YOUNG :   
  새롭게 생성한 객체가 대부분 위치, 대부분의 객체가 금방 접근 불가능 상태가 됨.  
  이 영역에서 객체가 사라질 때 Minor GC가 발생  
  \* OLD :   
  접근 불가능 상태로 되지 않고 YOUNG 영역에서 살아남은 객체가 복사되는 공간, YOUNG 영역보다 크게 할당, 크기가 크기 때문에 YOUNG 영역보다 GC가 적게 발생  
  이 영역에서 객체가 사라질 때 Major GC가 발생  
  \* PERM :  
  Method Area라고 함. 객체나 억류된 문자열 정보를 저장  
  객체가 사라질 때 Major GC가 발생

🏵 Minor GC

Young 영역은 Eden과 Survive1, 2로 나뉨

Eden 영역은 처음 객체가 생성될 때 상주,

Eden 영역에 객체가 가득 차면 첫 Garbage Collection 발생

Minor GC가 첫 번째 발생 시 Eden에서 살아남을 경우 Survive1로 이동, Eden Clear

Eden 영역과 Survive1 영역이 가득 차면 Minor GC가 두 번째 발생, Eden 영역에서 생성된 객체와 Survive1 영역에 있는 객체 중 참조되고 있는 객체 검사

참조되고 있는 객체를 Survivor2 영역으로 이동, Eden/Survive1 Clear

세 번째 Minor GC 발생 시, Eden가 Survive1의 오래된 객체들을 Old 영역으로 이동

🡺 Minor GC는 자주 발생 -> GC 소요되는 시간이 짧은 알고리즘   
이 방법은 속도가 빠르고 작은 크기의 메모리를 Collection 하는데 효과적.

🏵 Major GC

Mark : OLD 영역에 있는 참조하지 않는 객체들을 표시

Sweep : OLD 영역에 있는 참조하지 않는 객체를 삭제

Compact : 삭제되어 확보한 메모리가 부분부분 빈 공간을 생성하면 빈 공간을 매꾸기 위해 메모리 재구성 🡪 디스크 조각모음과 같은 원리

Mark는 전체 객체들의 레퍼런스를 확인하면서 레퍼런스와 연결되지 않은 객체들을 표시  
사용되지 않은 객체들이 Mark된 후 Mark된 객체를 삭제  
Compact는 Mark된 후 삭제되어 생기는 메모리를 객체로 매꿈.