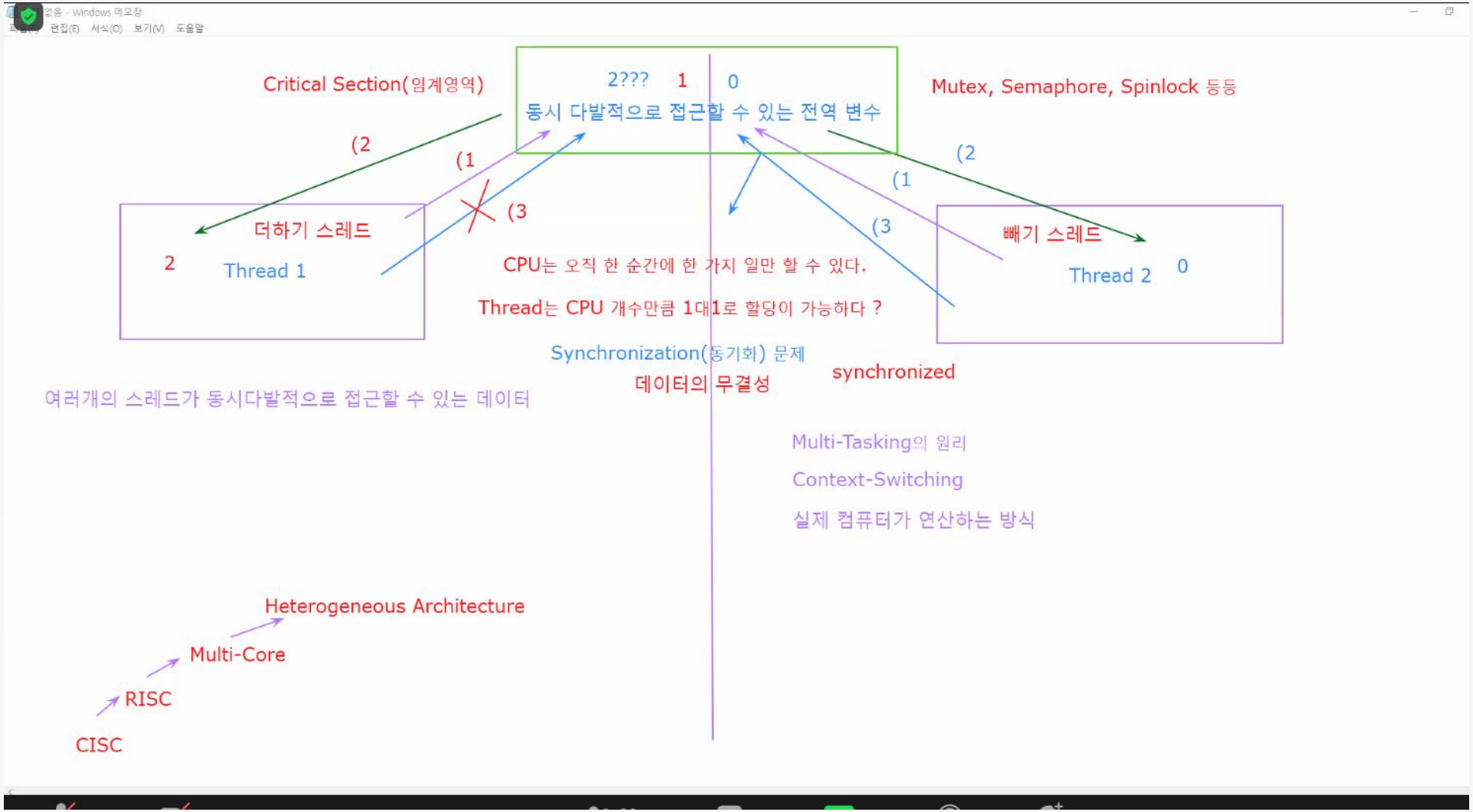
6월 3일 복습&퀴즈

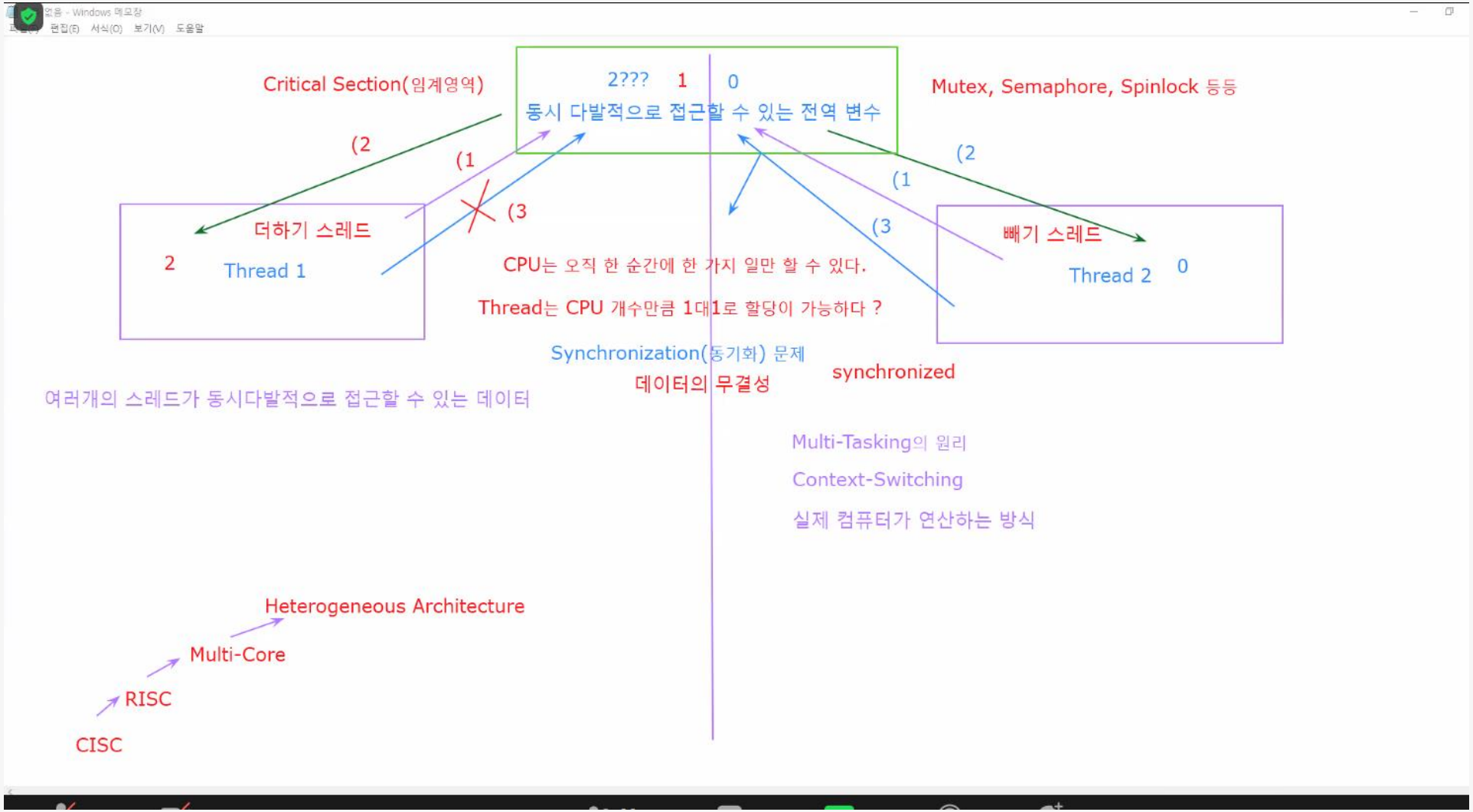
이태양


```
class Car implements Runnable {

    String name;
    private int sleepTime;
    private final static Random generator = new Random();

    public Car(String name) {
        this.name = name;
        sleepTime = generator.nextInt( bound: 3000) + 3000;
    }
    @Override
    public void run() {
        // try라는 키워드를 적는 경우는 I/O 나 특정한 이벤트,
        // 인터럽트 등등이 발생하는 경우에 작성하게 됨
        // 이 녀석은 에러를 유발할 수도 있다! 를 암시함
        try {
            Thread.sleep(sleepTime);
        } catch (InterruptedException e) {
            System.out.println("출력도 안 될 것이고 에러가 발생할 일이 없습니다!");
        }
        System.out.println(name + "이 경주를 완료하였습니다!");
    }
}
public class ThreadTest {
    public static void main(String[] args) {
        Thread t1 = new Thread(new Car( name: "Ferrari"));
        Thread t2 = new Thread(new Car( name: "Spyder 918"));
        Thread t3 = new Thread(new Car( name: "Maserati MC20"));

        t1.start();
        t2.start();
        t3.start();
```

스레드 시간을 0에서3초까지 랜덤할당해서 출력되는걸 볼 수 있다. Run()을 무조건 구동해야해서 run()을 꼭 기억하기!



임계영역 : 두 개의 스레드가 접근하지못하게 해줌
한번에 하나의 작업만 할 수 있다



임계영역 : 두 개의 스레드가 접근하지못하게 해줌
한번에 하나의 작업만 할 수 있다
데이터 무결성 : **데이터**의 정확성과 일관성을 유지하고 보증하는 것

ex) 컨텍스트 스위칭 도입이 없는 상태
Thread 1
mov eax, 3 // eax = 3 (1)
mov ebx, 4 // ebx = 4 (4)
add eax, ebx // eax = eax + ebx ---> (5) 7 + 4 = 11 계산이 틀려짐!
Thread 2
Mov eax, 7 // eax = 7 (2)
Mov ebx, 2 // ebx = 2 (3)
Add eax, ebx // eax = eax + ebx
Ex) 컨텍스트 스위칭을 도입한 상태
Thread 1
Mov eax, 3 // eax = 3 (1) ----> 제어권을 넘기기 전에 현재 하드웨어 레지스터 정보를 메모리에 저장함
(이 정보는 운영체제가 관리하고 있어서 찾을 수 있음)
mov ebx, 4 // ebx = 4 (4) <--- 다시 돌아와서 이전에 저장한 정보를 메모리에서 찾아서 복원함
add eax, ebx // eax = eax + ebx ---> ??? (5) 3 + 4 = 7 (데이터의 무결성을 보장함)
Thread 2
mov eax, 7 // eax = 7 (2)
mov ebx, 2 // ebx = 2 (3) ----> 자신의 제어권이 역시 넘어가기 전에 하드웨어 레지스터 정보를 백업함
add eax, ebx // eax = eax + ebx
Mutex vs Spinlock의 차이점이 뭡니까 ?
컨텍스트 스위칭이 도입이 되어있냐 안되어있냐의 차이가 있다
Spinlock같은경우는 덧셈뺄셈같은 간단한 처리를 할 때빠르게 처리할 수 있고
복잡할수록 Mutex가 더 좋다