

1. 프로세스 동기화(synchronization)에 대한 설명으로 가장 옳은 것은?
  - 1) 여러 프로세스가 공유 자원에 접근할 때, 병렬적(parallel)인 처리를 할 때는 항상 동기화가 필요하지만, 병행적(concurrent)인 처리를 할 때는 항상 동기화가 필요하지는 않다.
  - 2) 생산자-소비자 문제를 두 개의 프로세스가 shared memory로 처리할 때는 동기화를 해 주어야 하지만, 두 개의 쓰레드가 같은 주소 공간에서 buffer를 저장할 때는 동기화가 필요하지 않다.
  - 3) 여러 쓰레드가 어떤 공유하는 변수에 접근하여 값을 변경하지 않고 읽기만 하는 경우에는 경쟁 상황이 발생하지 않으므로 동기화를 해 줄 필요가 없다.
  - 4) 프로세스 동기화(synchronization)는 프로세스가 공유하는 자원에 접근하는 일련의 순서를 정하도록 한다. 따라서 공유하는 프로세스들이 항상 같은 순서로 순차적으로 자원에 접근할 수 있게 하여 경쟁 상황을 방지한다.
2. 임계영역(Critical Section)에 대한 설명으로 가장 틀린 것은?
  - 1) critical-section은 어떤 프로세스의 코드 영역 중에서 여러 프로세스가 공유하는 자원에 접근하는 코드 영역을 말하고, entry-section과 exit-section 사이에 위치한다.
  - 2) entry-section은 임계 영역에 진입하기 위한 권한을 획득하는 코드 영역을 말하고, 항상 critical-section 이전에 위치해야 한다.
  - 3) exit-section은 임계 영역을 빠져나와서 권한을 반환하는 코드 영역을 말하고, 항상 critical-section 이후에 위치해야 한다.
  - 4) remainder-section은 entry-, exit-, critical-section이 아닌 코드 영역을 말하고, 항상 exit-section 이후에 와야 한다.
3. 임계 영역 문제 (Critical-Section Problem)에 대한 솔루션이 해결해야 할 요구사항에 포함하지 않아도 되는 것은?
  - 1) Mutual Exclusion
  - 2) Progress (No deadlock)
  - 3) Bounded Waiting (No starvation)
  - 4) Scalability

4. 임계 영역 문제 (Critical Section Problem)의 해결책에 대한 설명으로 가장 옳은 것은?

- 1) Single-core 시스템에서는 단순히 인터럽트를 방지하는 것으로 임계영역 문제를 해결할 수 없다.
- 2) 비선점형(nonpreemptive) 커널에서는 경쟁 상황이 발생하지 않으므로 임계영역 문제를 고려할 필요가 없다.
- 3) 피터슨 알고리즘은 상호 배제 문제를 확실하게 해결했지만, 데드락과 기아 문제를 해결하기 위한 하드웨어적인 지원이 필요하다.
- 4) 원자적 변수(Atomic Variable)는 공유 변수에 대한 접근을 하는 명령어(instruction)를 하드웨어적으로 만들어 지원해 주는 임계 영역 문제 해결책이다.

5. 아래와 같이 피터슨 알고리즘을 구현했을 때, A, B, C, D에 들어갈 값으로 잘못 짚지어진 것을 모두 고르시오.

```
while (true) {  
    /* entry section */  
    flag[0] = true;  
    turn = A;  
    while (flag[B] && turn == C)  
        ;  
  
    /* critical section */  
    sum++;  
  
    /* exit section */  
    flag[D] = false;  
  
    /* remainder section */  
}
```

- 1) A = 0
  - 2) B = 1
  - 3) C = 0
  - 4) D = 1
6. 수업시간에 다룬 producer-consumer 예제에서, producer가 두 개의 쓰레드로 실행되고, consumer가 두 개의 쓰레드로 실행되었다고 가정해 보자. 만약 count의 값이 현재 5인 상태에서, 네 개의 쓰레드가 한 번씩 concurrent하게 실행되었다면, 다음 중 최종 count의 값을 가능한 값을 모두 고르시오.
- 1) 3
  - 2) 4
  - 3) 5
  - 4) 6
  - 5) 7

7. 하드웨어 솔루션으로 임계구역 문제를 해결할 때, 이에 대한 설명으로 가장 틀린 것은?
- 1) 임계영역 문제를 해결하기 위한 하드웨어 솔루션은 atomicity(원자성)를 보장하는 명령어를 제공한다.
  - 2) test\_and\_set 명령어는 불린 변수인 lock을 이용한다.
  - 3) compare\_and\_swap 명령어는 전역 변수인 lock을 이용한다.
  - 4) atomic\_variable 명령어는 변수에 대한 접근을 제어하는 하드웨어 명령어
8. 임계구역 문제에 대한 설명으로 가장 틀린 것은?
- 1) 경쟁 상황(race condition)이 발생할 수 있는 코드 영역을 임계구역(critical section)이라 한다.
  - 2) 피터슨 알고리즘은 임계구역 문제에 대한 소프트웨어 해결책이고, compare\_and\_swap은 하드웨어 해결책이다.
  - 3) atomic variable을 이용하여 생산자-소비자 문제의 동기화를 해결할 수 있다.
  - 4) Java에서 Peterson 알고리즘을 그대로 구현하면 data inconsistency가 발생하지 않는다.

**Answers** (indended by the Question Provider):

- 1) 3
- 2) 4
- 3) 4
- 4) 2
- 5) 1, 3, 4
- 6) 3, 4, 5, 6, 7
- 7) 4
- 8) 4