

운영체제론 기말고사'20

소프트웨어학부

2020년 6월 18일

1. [50 pts] 다음 글을 읽고 빈 칸에 알맞는 말을 채우거나 선택하라.

- (a) 자바는 CAS (Compare And Swap) 명령어를 지원하지 않는다. (참/거짓)이다.
- (b) 자바 모니터는 조건변수를 지원하지 않는다. (참/거짓)이다.
- (c) 스레드가 이미 소유한 락을 재소유하기 위해 시도해도 교착상태가 발생하지 않는 락을 _____(이)라고 부른다.
- (d) Deadlock (prevention / avoidance / detection / ignorance) 방식은 프로세스가 앞으로 어떤 자원이 얼마나 필요한 지에 대한 사전 정보를 이용한다.
- (e) 대부분의 OS는 deadlock (prevention / avoidance / detection / ignorance) 방식을 사용한다.
- (f) 식사하는 철학자 문제를 해결하기 위해 젓가락을 두 개 다 집을 수 있을 때만 집는 것은 교착상태가 일어나기 위한 4가지 필요조건 가운데 _____ 조건을 막기 위함이다.
- (g) 자원의 고유번호에 따라 오름차순으로 요청하게 하는 것은 교착상태가 일어나기 위한 4가지 필요조건 가운데 _____ 조건을 막기 위함이다.
- (h) 자원할당 그래프에서 사이클이 존재하면 교착상태가 일어났음을 의미한다. (참/거짓)이다.
- (i) TLB는 캐시의 일종으로 병렬검색이 가능한 _____ 메모리이다.
- (j) 인텔 x86 아키텍처는 현재 _____ 비트를 주소로 사용하고 있다.
- (k) 해시 페이지 테이블의 연결 리스트에 있는 항목에는 _____ 정보가 저장되어 있다.
- (l) 페이지의 크기가 8 KB이고 32 비트 주소체계를 사용하는 시스템이 1 GB (물리적인) 메모리를 가지고 있다면, 역페이지 (inverted page) 테이블의 엔트리는 _____ 개가 필요하다.
- (m) 페이지 테이블이 메모리에 있는 페이징 시스템에서 메모리 접근시간이 50 나노초이고, TLB 히트율이 75%이다. TLB 조회 시간을 무시하면, 이 시스템의 유효 메모리 접근시간은 _____ 나노초이다.
- (n) FIFO, LFU, MFU, LRU, OPT 중에서 참조열 S 의 페이지 폴트 수와 역참조열 S^R 의 페이지 폴트 수가 동일한 알고리즘은 _____ 이다.
- (o) 프레임의 수를 늘렸는데 오히려 페이지 폴트가 증가하는 현상을 _____(이)라고 부른다.
- (p) N 개의 프레임을 사용했을 때 남아있는 페이지의 집합이 $N + 1$ 개의 프레임을 사용했을 때 남아있는 페이지 집합의 부분집합을 항상 만족할 때, 이런 페이지 교체 알고리즘을 _____(이)라고 부른다.
- (q) 2차 기회 페이지 교체 (second-chance page replacement) 알고리즘에서 교체 대상이 되는 페이지의 참조비트가 0이면 _____ 하고, 1이면 _____ 한다.

- (r) 리눅스에서 마이너 페이지 폴트가 메이저 페이지 폴트보다 월등히 높은 이유는 _____을(를) 많이 활용하고 있다는 증거이다.
 - (s) 스레드의 상태를 new, ready, running, waiting, terminated로 구분하자. 실행 중인 스레드가 TLB miss가 발생해서 페이지 테이블에서 물리 주소를 얻었다면 스레드는 실행 상태에서 다른 상태로 변환되나? 만일 그렇다면 어떤 상태로 변환되나?
 - (t) 이번에는 실행 중인 스레드에서 페이지 폴트가 발생하면 스레드는 실행 상태에서 다른 상태로 변환되나? 만일 그렇다면 어떤 상태로 변환되나?
 - (u) 커널 메모리 할당 방식으로 SLAB을 사용하는 장점은 _____와(과) _____이다.
 - (v) _____이(가) 발생하는 이유는 프로세스에 할당된 프레임의 수가 워킹셋 (working set)을 수용할 만큼 충분하지 못해서 지속적으로 페이지 스와핑이 일어나기 때문이다.
 - (w) 모바일 운영체제에서는 메모리가 부족할 때 스와핑 대신에 _____을(를) 사용한다.
 - (x) N 개의 디스크가 RAID5에 묶여 있을 때, 저장된 어떤 블록의 일부를 변경하려면 패러티 블록의 변경을 포함해서 모두 _____번의 블록 읽기와 _____번의 블록 쓰기가 필요하다.
 - (y) NVM 장치에서 하나의 쓰기가 여러 개의 읽기와 쓰기를 유발하는 현상을 _____(이)라고 한다.
2. [7 pts] 다음은 유한버퍼 (bounded buffer) 문제를 자바 모니터를 사용하여 구현한 것이다. 클래스 속성을 참조하여 `remove` 메소드를 완성하라.

```
public class BoundedBuffer<E>
{
    private static final int BUFFER_SIZE = 5;

    private int count, in, out;
    private E[] buffer;

    public BoundedBuffer() {
        count = 0;
        in = 0;
        out = 0;
        buffer = (E[]) new Object[BUFFER_SIZE];
    }

    /* Producers call this method */
    public synchronized void insert(E item) {

    }

    /* Consumers call this method */
    public synchronized E remove() {

    }
}
```

3. [6 pts] 프로세스 P_1 의 주기는 $p_1 = 50$ 이고, CPU 버스트는 $t_1 = 25$ 이다. 프로세스 P_2 의 주기는 $p_2 = 80$ 이고, CPU 버스트는 $t_2 = 35$ 이다.
- (a) 이 두 프로세스가 RM (Rate Monotonic) 스케줄링으로 스케줄이 가능한가? 간트 차트를 사용하여 보여라.

- (b) 이 두 프로세스가 EDF (Earliest Deadline First) 스케줄링으로 스케줄이 가능한가? 간트 차트를 사용하여 보여라.
4. [7 pts] 다음 표는 어떤 시스템의 자원 할당 현황을 나타낸 것이다. Banker의 알고리즘을 사용하여 물음에 답하라.

	<u>Allocation</u>	<u>Max</u>	<u>Available</u>
	A B C	A B C	A B C
T_0	0 1 0	7 5 3	3 3 2
T_1	2 0 0	3 2 2	
T_2	3 0 2	9 0 2	
T_3	2 1 1	2 2 2	
T_4	0 0 2	4 3 3	

- (a) 현재 시스템이 안전한 상태에 있음을 보여라.
- (b) T_0 가 (0, 2, 0)를 요청했다면, 시스템은 이 요청을 받아들일 수 있나?
- (c) T_1 이 (1, 0, 2)를 요청했다면, 시스템은 이 요청을 받아들일 수 있나?
5. [4 pts] 프로그램이 어떤 메모리 주소를 참조했다고 가정하자. 다음의 경우가 발생할 수 있으면 사례를 들고, 불가능한 일이면 그 이유를 간단히 설명하라.
- TLB miss with no page fault
 - TLB miss with page fault
 - TLB hit with no page fault
 - TLB hit with page fault
6. [6 pts] 다음에 주어진 참조열을 프레임이 4개인 LRU와 OPT 페이지 교체 알고리즘에 적용했을 때 각각 몇 개의 페이지 폴트가 발생하는지 그 과정을 보여라.
- 7, 0, 1, 2, 0, 3, 0, 4, 2, 3, 0, 3, 0, 3, 2, 1, 2, 0, 1, 7, 0, 1
7. [6 pts] 다음 표는 16 비트 주소체계를 사용하고 페이지의 크기가 4KB인 어떤 시스템의 페이지 테이블이다. ‘-’ 문자는 해당 페이지가 현재 프레임에 할당되지 않았음을 나타낸다. 프레임 넘버는 10진수로 표현되어 있다. 페이지가 조회되면 reference bit가 1로 설정된다.

Frame Number	Reference Bit
9	0
-	0
10	1
15	0
6	0
13	0
8	0
12	0
7	1
-	0
5	0
4	0
1	1
0	0
-	0
2	0

(a) 다음에 열거한 16진수 가상(논리) 주소를 물리 주소로 변환하라.

- 0x621C
- 0xBC1A

(b) 페이지 테이블에 따르면 페이지 폴트가 발생하는 가상 주소의 범위는 어떻게 되나?

(c) 앞 문제 (a)에서 열거한 주소가 조회된 후에 페이지 폴트가 발생하여 페이지 교체가 필요하게 되었다. LRU 근사치 페이지 교체 알고리즘을 사용한다면 교체 대상이 될 수 있는 페이지 프레임 후보는 어떤 것들인가?

8. [14 pts] 다음은 읽는자-쓰는자 문제를 해결하기 위한 의사코드(psuedo-code)로 읽는자를 선호하는 방식이다. 코드를 잘 읽고 물음에 답하라.

<pre>void *reader(void *arg) { while (ture) { wait(mutex); read_count++; if (read_count == 1) wait(rw_mutex); signal(mutex); ... /* reading is performed */ ... wait(mutex); read_count--; if (read_count == 0) signal(rw_mutex); signal(mutex); } }</pre>	<pre>void *writer(void *arg) { while (ture) { wait(rw_mutex); ... /* writing is performed */ ... signal(rw_mutex); } }</pre>
--	--

(a) 앞의 읽는자 선호 방식을 POSIX 조건변수를 사용하여 reader와 writer를 구현하라. POSIX API는 다음과 같다. 단, 메인 함수, 뮤텝스 락이나 조건변수에 대한 초기화는 생략해도 무방하다.

```
int pthread_mutex_lock(pthread_mutex_t *mutex);
int pthread_mutex_unlock(pthread_mutex_t *mutex);
int pthread_cond_wait(pthread_cond_t *restrict cond,
                      pthread_mutex_t *restrict mutex);
int pthread_cond_signal(pthread_cond_t *cond);
int pthread_cond_broadcast(pthread_cond_t *cond);
```

(b) POSIX 뮤텝스 락을 사용하여 공정한 읽는자-쓰는자 방식을 구현하라. 조건변수나 세마포를 사용해서는 안된다. 단, 메인 함수, 뮤텝스 락에 대한 초기화는 생략해도 무방하다.

HK