운영체제론 기말고사'22

소프트웨어학부

2022년 6월 14일

1. [30 pts] 다음 글을 읽고 물음에 답하라.

   (1) 자원에 순번을 정해서 오름차순으로 요청하는 것은 교착상태(deadlock)가 발생하기 위한 (충분/필요/필요충분)조건 가운데 (mutual exclusion/hold and wait/no preemption/circular wait)를 방지하기 위함이다.

   (2) 어떤 자원을 소유한 순위가 낮은 프로세스가 그 자원을 요청한 순위가 높은 프로세스의 우선순위를 할당받는 행위를 _____라 부른다.

   (3) 페이지의 크기가 16 KB이고 32 비트 주소체계를 사용하는 시스템이 있다. 단일 계층 페이지 테이블을 사용하고 방(entry)의 크기가 4 바이트이면 하나의 페이지 테이블을 저장하기 위한 공간은 _____ 바이트가 필요하다.

   (4) 페이지의 크기가 64 KB이고 64 비트 주소체계를 사용하는 시스템이 있다. 계층형 페이지 테이블을 사용한다면 적어도 몇 계층이 필요한가? 전체 64 비트 가운데 각 계층이 사용하는 비트의 길이는 얼마인가? 단, 각 계층마다 사용하는 테이블이나 디렉토리의 크기는 페이지 크기를 넘을 수 없다.

   (5) 문장 `z = x + y`를 실행할 때 페이지 폴트가 5번 일어날 수 있나? 만일 있다면 어떤 경우인가?

   (6) 페이징 시스템에서 메모리가 부족할 때 공간을 늘리기 위해 메모리 압축을 사용한다. 압축 대상이 되는 프레임은 어떤 것인가?

   (7) 프로세스에 할당된 프레임이 3개인 상태에서 LRU 페이지 교체 알고리즘을 모의실험하였다. 참조열 S를 조회한 후 프레임에 남아 있는 페이지가 {2,3,4}라면 프레임을 4개로 늘렸을 때 조회 후 프레임에 남아 있는 페이지의 집합이 될 수 없는 것을 모두 골라라. {1,2,3,4}, {1,2,3,5}, {1,3,4,5}, {1,2,4,5}, {2,3,4,5}.

   (8) 참조비트(reference bit)만 사용하여 LRU 근사치 알고리즘을 수행하는 요구페이징(demand paging) 시스템에서 처음 불러온 페이지의 참조비트는 (0/1)이다. 선택하라. 이 참조비트의 값은 언제 어떤 상황에서 어떻게 변경되나?

   (9) 2차 기회 페이지 교체 (second-chance page replacement) 알고리즘에서 교체 대상이 되는 페이지의 참조비트가 0이면 _____ 하고, 1이면 _____ 한다.

   (10) 리눅스에서 프로그램을 처음 실행할 때보다 이어서 재실행할 때는 메모리에 그 내용이 남아 있어서 페이지 폴트의 수가 현격히 줄어든다. (참/거짓)이다.

2. [10 pts] 어떤 요구페이징 시스템에서 페이지 폴트가 발생했을 때, 빈 프레임이 있을 경우에는 8 msec 비용이 든다. 페이지 교체가 필요하여 교체되는 페이지를 저장공간에 쓸 경우에는 20 msec 비용이 들고, 그럴 필요가 없는 경우에는 8 msec 비용이 든다. 페이지 폴트가 일어나지 않으면 메모리 접근 비용은 200 nsec이다. 이 비용에는 TLB와 페이지 테이블 접근 비용까지 포함되어 있다. 페이지 폴트가 일어났을 때, 빈 프레임이 있을 확률은 10%이고, 없어서 기존의 페이지를 교체할 경우 교체되는 페이지가 변경되어 있을 확률은 70%이다. 페이지 폴트 비율이 0.0001일 때 이 시스템의 유효 접근 시간은 몇 nsec인가? 시간 단위에 주의한다. 1 msec는 10^6 nsec이다.

3. [10 pts] 임계구역을 보호하기 위해 전역변수 lock을 사용하여 스핀락을 구현하였다. 물음에 답하라.

```
int lock = 0; // 스레드간 공유하는 전역변수
...
while (lock != 0)
    /* do nothing */;
lock = 1;
// 임계구역 시작
...
// 임계구역 종료
lock = 0;
```

(a) 이 코드의 문제점에 대해서 논하라.

(b) atomic_compare_exchange_strong() 함수를 사용하여 위 코드를 올바르게 수정하라. 함수의 원형은 다음과 같다: _Bool atomic_compare_exchange_strong(_Atomic(T) *object, T *expected, T desired);

4. [10 pts] 정수형 원자변수 x의 값을 d만큼 증가시키는 원자함수를 구현하였다. 이 함수는 x를 d만큼 증가시키는 과정을 원자적으로 수행하며, 변경되기 전의 x 값을 리턴한다. 물음에 답하라.

```
int atomic_add(atomic_int *x, int d)
{
    int tmp = *x;

    *x += d;    // x를 d만큼 증가시키고,
    return tmp; // 증가시키기 전의 값을 리턴한다.
}
```

(a) 이 코드의 문제점에 대해서 논하라.

(b) ⋆ atomic_compare_exchange_strong() 함수를 사용하여 위 코드를 올바르게 수정하라. 단, 다른 함수나 추가변수를 사용할 수 없다.

5. [10 pts] 식사하는 철학자 문제를 다음과 같이 구현하였다. 사용된 뮤텍스락은 올바르게 초기화되었다고 가정한다. 물음에 답하라.

```
#define N 5
pthread_mutex_t chopstick[N];
...
while (true} {
    pthread_mutex_lock(&chopstick[i]);
    pthread_mutex_lock(&chopstick[(i+1)%N]);
    // 철학자 식사 중
    pthread_mutex_unlock(&chopstick[i]);
    pthread_mutex_unlock(&chopstick[(i+1)%N]);
    // 철학자 생각 중
}
...
```

(a) 이 코드는 교착상태에 빠질 수 있다. 그 시나리오를 제시하라.

(b) 교착상태에 빠지지 않게 위 코드를 올바르게 수정하라.

6. [10 pts] 흡연자 문제를 세마포를 사용하여 구현하였다. 지면이 좁아 종이만 가지고 있는 흡연자를 제외한 나머지 코드는 생략하였다. 물음에 답하라.

```
sem_t *tabacco, *paper, *matches, *done;
...
// 연초를 말 수 있는 종이만 가지고 있는 흡연자
void *paper_smoker(void *arg)
{
    while (true) {
        sem_wait(tabacco);
        sem_wait(matches);
        // 담배를 피운다.
        sem_post(done);
    }
}
...
```

(a) 앞 코드는 교착상태를 방지하기 위해 모든 흡연자는 연초, 종이, 성냥의 순서로 자원을 요청한다. 예를 들어, 성냥만 가지고 있는 흡연자는 순서에 따라 연초와 종이의 순으로 요청한다. 그럼에도 불구하고 교착상태가 발생한다. 그 이유는 무엇인가? 자원에 순번을 정해서 오름차순으로 요청하는 것이 틀렸다는 뜻인가?

(b) 교착상태에 빠지지 않는 방법을 제시하고, 그에 따라 위 함수를 올바르게 수정하라.

7. [10 pts] 다음은 읽는자-쓰는자 문제를 해결하기 위한 의사코드(psuedo-code)로 읽는자를 선호하는 방식이다. 코드를 잘 읽고 물음에 답하라.

```
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);
    }
}
```

```
void *writer(void *arg)
{
    while (ture) {
        wait(rw_mutex);
        ...
        /* writing is performed */
        ...
        signal(rw_mutex);
    }
}
```

위 알고리즘을 POSIX 조건변수 2개와 뮤텍스락 1개만을 사용하여 구현하라. 다만 다른 변수는 자유롭게 사용할 수 있다. 뮤텍스락이나 조건변수에 대한 초기화와 메인 함수는 생략한다.