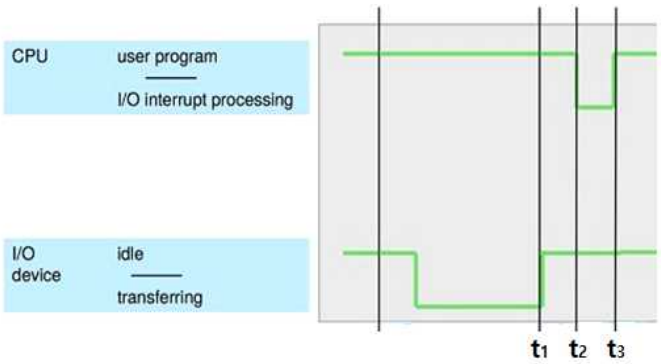
대학원 고급운영체제 종합시험 문제은행

2021년 3월 개정

※ 문제은행의 문제들은 기본 유형으로서, 출제되는 문제는 응용하여 재구성될 수 있음을 인지하기 바랍니다.

1. 다음 그림은 CPU와 I/O device 간 Timeline을 보여주고 있다. 그림을 통해 예측할 수 있는 작업 상황에 대해 설명하고, 시간 t_1, t_2, t_3에서 CPU와 I/O device 사이에 발생하는 이벤트 작업에 대해 매 시간별로 설명하시오.



2. 마이크로커널(microkernel) 구조와 모놀리틱커널(monolithic kernel) 구조의 차이점에 대해 서 커널 서브시스템(kernel subsystem)의 프로그램 실행 레벨과 보호영역(protection domain) 관점에서 서술하시오.

3. DMA(direct memory access)를 사용하여 CPU의 실행 부하(execution load)없이 고속 입출력 장치들을 사용하고자 한다. 이때 장치로의 메모리 연산이 완료되었음을 CPU가 알 수 있는 방법이 무엇이며, 그 방법과 트랩(trap)과의 차이에 대해서 서술하시오.

4. 임계구역 문제(Critical-section problem)의 해결책은 세 가지 요구사항인 Mutual exclusion, Progress, 그리고 Bounded waiting을 만족시켜야 하는데, 이러한 세 가지 요구사항에 대해 설명하시오.

5. Multiprocessor 환경에서 발생될 수 있는 문제 중 하나인 cache coherency에 대해 설명하시오.

6. CPU 스케줄링 알고리즘들(scheduling algorithm)인 FIFO(First In First Out), SJF(Shortest Job First), RR(Round Robin)의 스케줄링 방식에 대해 설명하고, 장단점을 비교 분석하시오.

7. 선점형 스케줄러(preemptive scheduler)와 비선점형 스케줄러(non-preemptive scheduler)를 설명하고, 응답성 및 예측 가능성 측면에서 비교 분석하시오.

8. Short-term scheduler와 Long-term scheduler의 차이점을 설명하고, (1)과 (2) 상황이 발생하는 경우, 스케줄러(scheduler)와 큐(queue) 관점에서 자세히 설명하시오.
   (1) If all processes in the ready queue are I/O bound
   (2) If all processes in the ready queue are CPU bound

9. SJF(Shortest Job First) 스케줄링 알고리즘은 실질적으로 구현하기가 어렵다. 그 이유를 설명하고 해결하기 위해 시도할 수 있는 방안에 대해 자세히 설명하시오.

10. Time quantum (혹은 time slice)을 설명하고, 태스크의 특성과 관련하여 time quantum의 크기와 스케줄러의 성능에 관한 연관 관계를 설명하시오.

11. 두 프로세스들 P_1, P_2의 periods가 p_1=50, p_2=100이고, processing time은 t_1=20, t_2=35이라고 하자. 이 때, 두 프로세스들이 실시간 CPU 스케줄링(real-time CPU scheduling) 기법들 중 하나인 rate-monotonic scheduling 기법으로 스케줄링될 때, 다음의 조건에 따른 수행 과정을 Gantt chart로 보이고 스케줄링이 적절히 되는지 여부를 설명하시오.
   (조건) When P_2 has a higher priority than P_1

12. 두 프로세스들 P_1, P_2의 periods가 p_1=50, p_2=100이고, processing time은 t_1=20, t_2=35이라고 하자. 이 때, 두 프로세스들이 실시간 CPU 스케줄링(real-time CPU scheduling) 기법들 중 하나인 rate-monotonic scheduling 기법으로 스케줄링될 때, 다음의 조건에 따른 수행 과정을 Gantt chart로 보이고 스케줄링이 적절히 되는지 여부를 설명하시오.
   (조건) When P_1 has a higher priority than P_2

13. 두 프로세스들 P_1, P_2의 periods가 p_1=50, p_2=80이고, processing time은 t_1=25, t_2=35이라고 하자. 이 때, 두 프로세스들이 실시간 CPU 스케줄링(real-time CPU scheduling) 기법들 중 하나인 earliest-deadline-first (EDF) scheduling 기법으로 스케줄링될 때, 다음의 조건에 따른 수행 과정을 Gantt chart로 보이고 스케줄링이 적절히 되는지 여부를 설명하시오.
   (조건) When P_1 has a higher priority than P_2

14. Interprocess Communication(IPC) 모델은 shared memory 방식과 message passing 방식으로 나누는데, 각 방식에 대해 설명하고 특징 및 장단점을 비교하시오.

15. 다음은 Eisenberg와 McGuire가 제안한 n개의 프로세스들에 대한 critical-section (**CS**) 문제의 해결방안으로서, 프로세스 P_i의 구조이다. 프로세스들이 공유하는 변수들이 다음과 같다고 하자.

```
enum pstate {idle, want-in, in-cs}
pstate flag[n];  /*  All the elements of flag are initially idle  */
int turn;    /* The initial value of turn is immaterial (between 0 and n-1) */
```

```
do
while (True)
   flag[i] := want-in ;
   j :=  turn ;
   while (j !=i )
       if (flag[j] != idle )
       then j := turn
       else j := (j+1) % n ;
   flag[i] := in-cs ;
   j := 0;
   while ((j<n) and (j==i or flag[j] !=in-cs)) j:=j+1;
   if (( j>=n) and (turn==i or flag[turn]==idle ))
       break;

          //    critical section (CS)

   j := (turn+1) % n ;
   while (flag[j]==idle)  j := (j+1) % n ;
   turn := j ;
   flag[i] := idle ;

          //    remainder section
while (True);
```

Critical-section 문제의 해결방안으로서, 세 가지 요구사항을 만족하는지 여부를 자세히 설명하시오.

16. 페이징 기법을 이용하는 가상 메모리 구조에서는 메모리에 해당 페이지가 없을 때, 페이지 교체를 통해 원하는 페이지를 메모리에 적재한 후 사용한다. 그러나 이런 페이지 교체가 자주 발생하게 되면, 프로세스의 처리 시간보다 메모리의 페이지 교체 시간이 더 길어지는 쓰레싱(thrashing) 문제가 발생할 수 있다. 이와 같은 문제의 원인과 해결 방안을 설명하시오.

17. 4 GB 가상메모리(virtual memory)를 갖는 시스템에서 페이지 크기(page size)가 1 MB 라고 할 때, 다음 물음에 답하시오.

(1) 페이지 테이블에 저장할 수 있는 엔트리의 수를 구하시오.

(2) 같은 프로세스를 처리하는 시스템 환경에서 페이지 크기(page size)가 4KB로 변경된다고 할 때, I/O time과 Internal fragmentation 측면에서 예측할 수 있는 오버헤드(overhead) 또는 장점에 대해 설명하시오.

18. 4개의 frames을 사용하는 메모리 시스템(memory system)에서 프로세스 수행을 위한 reference string이 다음과 같을 때, 다음 3가지 Page-replacement algorithms을 사용하여 교체되는 과정을 그림에 표현하고 발생한 페이지 부재(page faults) 수를 구하시오. (단, 교체 대상 페이지가 2개 이상인 경우에는 FIFO 기법을 적용함.)

(1) LRU replacement

1	2	3	4	5	3	4	1	6	7	8	7	8	9	7	8	9	5	4	5

(2) FIFO replacement

1	2	3	4	5	3	4	1	6	7	8	7	8	9	7	8	9	5	4	5

(3) Optimal replacement

1	2	3	4	5	3	4	1	6	7	8	7	8	9	7	8	9	5	4	5

19. 5개의 프로세스들 P_0, P_1, P_2, P_3, P_4에 할당된 자원의 수(Allocation), 작업 완료시까지 필요한 최대 자원의 수(Max), 그리고 현재 가용한 자원의 수(Available)가 다음과 같다고 하자. Deadlock handling methods 중 하나인 Banker's Algorithm을 사용하여 시스템이 안정 상태(safe state)인지 여부를 판단하시오.

	Allocation				Max				Available			
	A	**B**	**C**	**D**	**A**	**B**	**C**	**D**	**A**	**B**	**C**	**D**
P_0	0	0	1	2	0	0	1	2	1	5	2	0
P_1	1	0	0	0	1	7	5	0				
P_2	1	3	5	4	2	3	5	6				
P_3	0	6	3	2	0	6	5	2				
P_4	0	0	1	4	0	6	5	6				

안정 상태이면 처리 순서(safe sequence)를 구하고, 안정 상태가 아니라면 그 이유를 설명하시오. (단, safe sequence를 구할 수 있다면, 매 단계를 자세히 서술.)

20. 유닉스 I-node가 10개의 직접 접근 블록과 각 1개씩의 1차(single), 2차(double) 간접 접근 블록(indirect block)까지 활용한다고 할 때, 한 파일이 표현할 수 있는 최대 용량을 계산하시오. 단, 하나의 디스크 블록은 1KB 이며, 하나의 디스크 블록 주소는 4 Bytes이다. (계산기 불필요 최종 결과는 수식으로 표현 가능)

21. 버퍼 캐시를 LRU 정책과 FIFO 정책 두 가지 방식을 사용한다고 할 때, 아래 액세스 패턴에 대해 총 액세스 타임을 계산하시오. 액세스는 블록 단위로 이루어지며, 버퍼 캐시는 총 3개의 블록을 저장할 수 있다고 가정한다. 버퍼 캐시에서의 액세스 타임은 0.1ms, 그 외의 경우는 10ms로 계산할 것.

액세스 패턴: A, B, C, D, A, E, C, B, A, D