컴퓨터정보통신대학원 종합시험 기출문제
과목 : 운영체제

1. CPU 스케줄링 기법인 RR(Round Robin) 알고리즘에 대한 물음에 답하시오.

(1-1) RR 알고리즘에 대한 스케줄링 방식에 대해 설명하고, RR 알고리즘에서 설정하는 time quantum 크기 측면에서 FIFO(First In First Out) 알고리즘과 비교 설명하시오.

(1-2) RR 알고리즘에서 time quantum 크기가 작을수록 발생할 수 있는 오버헤드 측면에서 큰 경우와 비교하여 장단점을 설명하시오.

2. Page replacement 방식 중 하나인 LRU(least-recently-used) 기법에 대한 물음에 답하시오.

(2-1) LRU 기법에서 교체할 page를 결정하기 위한 정책을 설명하고, 그 구현 방법 2가지를 자세히 설명하시오

(2-1) 정확한 하나의 LRU page를 선택하는 방식이 아닌, LRU-Approximation page를 결정하기 위한 구현 방법 2가지를 자세히 설명하시오.

3. 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 스래싱(Thrashing) 문제가 발생할 수 있다. 이와 같은 문제의 원인과 해결책을 쓰시오.

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

4. 멀티프로그래밍만을 지원하는 운영체제 A가 있다. A에서 수행 가능한 형태로 적합하지 않은 것은?

① 다수의 응용프로그램을 메모리에 적재하여 실행할 수 있다.
② 응용프로그램이 높은 CPU 활용도(utilization)를 요구하는 경우 효과적이다.
③ 응용프로그램의 입출력 요청이 완료되기까지 대기해야 하는 시간을 최대한 활용할 수 있다.
④ 응용프로그램이 짧은 응답시간(response time)을 요구하는 경우 효과적이다.

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

6. 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

7. CPU scheduling algorithm들인 First In First Out, Shortest Job First, Round Robin을 비교하고 장단점을 분석하시오.

8. 세마포어(semaphore)에 관한 설명 중 틀린 것은?

① 상호 배제 문제를 해결하기 위하여 사용된다.

② 정수의 변수로 양의 값만을 가진다.

③ 여러 개의 프로세스가 동시에 그 값을 수정하지 못한다.

④ 세마포어에 대한 연산을 처리 도중에 인터럽트 되어서는 안 된다.

9. 다음 자기 디스크의 접근 시간에서 시간이 가장 많이 소요되는 것은?

① 탐색 시간 ② 회전 지연 시간

③ 헤드 활성화 시간 ④ 전송 시간

10. 페이지 부재(page fault)를 처리하는 순서로 올바른 것은?

a. 운영체제에서 트랩(trap)이 발생한다.
b. 페이지 테이블을 재조정한다.
c. 현재 사용자 레지스터와 프로그램의 상태를 저장한다.
d. 사용 가능한 메모리 프레임을 프레임 리스트에서 찾는다.
e. 명령어 수행을 계속한다.
f. 디스크와 같은 backing store에 있는 페이지를 물리 메모리로 가져와 적재한다.

① a – b – c – d – e – f

② a – c – f – b – d – e

③ a – c – d – f – b – e

④ c – f – b – a – d – e

⑤ e – a – d – b – c – f

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

12. 선점형 스케줄러와 비선점형 스케줄러를 설명하고, 응답성, 예측 가능성 측면에서 비교/분석 하시오.

13. 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 세 개의 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

14. 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.

15. 5개의 프로세스들 P0, P1, P2, P3, P4에 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.)

16. 아래와 같은 프로그램을 실행할 때, 첫 부모 프로세스를 포함해서 몇 개의 프로세스가 생성되는가?

```
#include  <stdio.h>
#include  <unistd.h>
int  main()
{
      fork();
      fork();
      fork();
      return  0;
}
```

17. Critical-section problem의 해결책은 Mutual exclusion, Progress 그리고 Bounded waiting의 요구사항을 만족 시켜야 한다. 각 3가지 요구사항을 설명 하시오.

18. 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)

주의: 계산 과정을 보이시오. 결과만 쓴 답안은 점수를 받지 못함.

19. 페이징 기법을 이용한 시스템에서 페이지의 크기가 작을 경우와 클 경우에 나타나는 영향을 기술하시오.

20. 임계구역(critical-section) 문제의 해결을 위한 3가지 요구사항은 Mutual exclusion, Progress, Bounded waiting이다. 다음에 주어진 Algorithm 1과 2가 두 프로세스에 대한 임계구역 문제 해결을 위한 3가지 요구사항을 만족하는지 여부를 판단하시오. 그리고 만족 여부에 대해 자세히 설명하시오. (설명없이 만족여부만 판단하는 경우 0점)

boolean flag[2]; /* initially **false** */

int turn; /* **i** or **j** */

Algorithm 1	Algorithm 2

repeat
flag[i] = true;
while (flag[j]) ;
Critical Section
flag[i] = false;
Remainder Section
until false;
	repeat
	flag[i] = true;
	turn = j;
	while (flag[j] && turn == j) ;
	Critical Section
	flag[i] = false;
	Remainder Section
	until false;

21. CPU 스케줄링 알고리즘들 중 FIFO(First In First Out)와 RR(Round Robin) 기법에 대해 비교 설명하시오. 그리고 이러한 두 알고리즘 간의 관계를 설명하시오.

22. 가상메모리 구조에서 페이징 기법을 사용하는 경우 쓰레싱(thrashing) 문제가 발생할 수 있다. 이 문제가 발생할 수 있는 상황에 대해 자세히 설명하고, 해결 방안에 대해 서술하시오.

23. 가상메모리(virtual memory)의 크기가 256 MB인 시스템에서, 가상메모리의 크기는 변동없이 페이지 크기가 4,096 bytes에서 256 bytes 로 변경되었다고 가정하자. 이 때, 논리 주소 공간에서 페이지크기를 의미하는 영역의 크기(bits의 개수)는 어떻게 변화하는지 서술하시오. 또한, 이와 같이 페이지 크기의 변화가 내부단편화(internal fragmentation) 및 전체 입출력시간에 미치는 영향에 대해 설명하시오.

24. Realtime operating system의 스케줄링 정책인 priority-based 방식으로 프로세스들 X, Y를 스케줄링한다고 가정하자. X와 Y의 periods인 p_x=50, p_y=80이고 processing times인 t_x=30, t_y=25 일 때, 다음 물음에 답하시오.

(24-1) 프로세스들 X, Y의 periods와 processing time을 사용하여 프로세스들 X, Y, 그리고 전체 CPU Utilization을 구하시오

(24-2) X, Y가 rate-monotonic 스케줄링 방식을 사용하여 스케줄링된다고 할 때, 수행되는 과정을 Gantt chart로 표현하고 스케줄링 가능 여부를 설명하시오. (스케줄링 정책은 "the shorter the period, the higher the priority" 라 가정)

(24-3) X, Y가 EDF(earliest-deadline-first) 스케줄링 방식을 사용하여 스케줄링이 가능한지 Gantt chart를 그려서 그 과정을 자세히 설명하시오.

25. Page replacement 기법에 대한 물음에 답하시오.

(25-1) Page replacement가 필요한 상황에 대해 설명하고, Page들의 교체 (replacement)가 수행되는 과정을 자세히 서술하시오.

(25-2) Page replacement를 위한 LRU(least-recently-used) 및 Optimal 기법 각각의 특징에 대해 서술하고 두 기법을 비교 설명하시오.

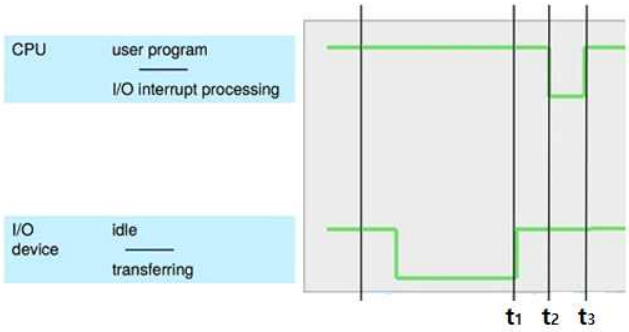
26. CPU스케줄링 기법에 대한 물음에 답하시오.

(26-1) Round robin 기법에 대해 설명하고, time quantum의 크기에 따라 스케줄링 성능에 영향을 주는 부분에 대해 서술하시오.

(26-2) Multi-level Queue 스케줄링 기법의 특징을 설명하고 그 구현 방법에 대해 서술하시오.

(26-3) Shortest Job First 와 Priority 기법간의 관계를 설명하시오.

27. 다음 그림은 CPU와 I/O device 간 Timeline을 표현하고 있다. 물음에 답하시오.



(27-1) 시간 t_1에서 I/O device의 상태가 바뀌는데, 그 상황을 I/O device 측면에서 설명하시오.

(27-2) 시간 t_2에서 CPU의 상태 변화 이유를 설명하고, CPU가 처리하는 과정에 대해 자세히 서술하시오.

(27-3) 시간 t_3에서 CPU가 수행하는 과정에 대해 자세히 설명하시오.

28. 운영체제의 스케줄러 유형은 크게 두 가지로 나눌 수 있는데, 하나는 단기 스케줄러(Short-term scheduler)이고 다른 하나는 장기 스케줄러(Long-term scheduler)이다. 관련된 물음에 대해 답하시오. [35점]

(28-1) 이러한 두 가지 스케줄러의 역할과 기능 측면에서 비교 설명하시오.

(28-2) 큐(ready queue)에서 대기하는 프로세스들의 유형이 "I/O bound"인 경우, 스케줄러의 작업 수행에 있어서 그 현상을 설명하시오.

(28-3) 큐(ready queue)에서 대기하는 프로세스들의 유형이 "CPU bound"인 경우, 스케줄러의 작업 수행에 있어서 그 현상을 설명하시오.

29. 페이징(paging) 기반 가상메모리 구조에서 처리에 필요한 페이지가 메모리에 로딩되어 있지 않으면, 페이지 폴트(page fault)가 발생한다. 이에 따라 최악의 경우 발생할 수 있는 문제인 쓰레싱(thrashing)에 관련된 물음에 대해 답하시오. [30점]

(29-1) 쓰레싱의 발생 원인에 대해 구체적으로 서술하시오.

(29-2) 쓰레싱 문제를 해결하기 위한 방안에 대해 설명하시오.

30. 교착상태(deadlock)와 관련된 물음에 답하시오. [35점]

(30-1) 교착상태 발생에 필요한 4가지 시스템 조건(conditions)에 대해 설명하시오.

(30-2) 교착상태를 방지(prevention)할 수 있는 방안에 대해 4가지 시스템 조건 측면에서 설명하시오.

(30-3) 시스템에서 5개의 프로세스들 P_0, P_1, P_2, P_3, P_4가 자원들 A(13개), B(10개), C(6개), D(9개)을 가지고 실행된다고 할 때, Banker's Algorithm을 사용하여 다음 물음에 답하시오.

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

모든 프로세스들이 작업을 완료할 수 있는 안정 상태(safe state)인지 여부를 판단하시오. 안정 상태이면 처리 순서(safe sequence)를 구하고, 안정 상태가 아니라면 그 이유를 설명하시오. (단, safe sequence를 구하는 경우, 매 단계를 자세히 서술하시오.)

31. CPU 스케줄링을 위한 FCFS(First-Come First-Served), SJF(Shortest Job First), Priority, RR(Round Robin) 알고리즘과 관련된 물음에 답하시오. [50점]

(31-1) FCFS와 RR 알고리즘의 특징을 서술하고, 두 알고리즘들간 관계를 설명하시오.

(31-2) SJF 알고리즘의 이론적 성능이 가장 좋다고 알려져 있지만, SJF에서 구현시 고려해야 할 문제점에 대해 서술하시오. 또한, 이 문제를 해결하기 위한 이론적 해결 방안에 대해 설명하시오.

(31-3) Priority와 SJF 알고리즘들간 관계를 설명하시오.

(31-4) RR 알고리즘에서 time quantum의 크기와 CPU 이용률간 관련성을 설명하시오.

32. 알고리즘은 임의의 프로세스 P_i의 임계구역(critical-section; CS) 문제에 대한 해결방법으로서, n개의 프로세스들이 공유하는 변수들은 다음과 같다. [50점]

```
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);
```

(32-1) 프로세스 P_i가 임계구역을 실행중 일 때, 다른 프로세스들의 임계구역 진입이 불가능함을 알고리즘 코드 내용 기반으로 설명하시오.

(32-2) 프로세스 P_i가 임계구역 실행 종료 후, 임계구역 진입을 기다리는 프로세스들 중 어느 프로세스가 진입할 것인가를 결정하는 방법을 코드 내용 기반으로 설명하시오.

(32-3) 임계구역 문제의 요구사항들인 Mutual exclusion, Progress, Bounded waiting의 개념에 대해 설명하시오.