

[1] Suspended state가 필요한 이유 중 틀린 것은? (하나를 선택하세요.)

1. 디스크와의 input/output 횟수를 줄이기 위해
2. Multiprogramming degree를 조정하기 위해
3. 메모리를 효율적으로 사용하기 위해
4. 프로세스의 실행과정이 의심될 때

정답: 1

[2] CPU가 하나 있는 시스템의 부하 조절(Load control)은 기본적으로 다중 프로그래밍 (Multiprogramming)의 도수(Degree)조절에 있다. 다음 중 이런 경우의 부하 조절과 상관없는 이론은?

1. 50% 이론 p183 부재 간격의 평균값(MTBF)과 부재 처리에 걸리는 시간의 평균 값을 같도록
2. L = S 이론 P183 페이징 장치의 활용도를 50%가 되도록

3. Fair share group p66 CPU 스케줄링

4. Working set 이론 p181 다중프로그래밍의 정도를 올리는 것은 working set이 메모리에 유지될 수 있느냐가 기준
- 정답: 3

[3] 아래 표와 같이 서로 다른 CPU 요구시간과 함께 네 개의 job이 시스템에 도착하였다. job a의 도착과 동시에 시스템이 수행 된다고 할 때 Shortest Remaining Time first (SRT)로 스케줄 하였을 때의 평균 대기시간 (Average Waiting Time)을 고르세요.

job	도착시간	CPU 요구시간
a	0	6
b	1	5
c	2	3
d	3	7

1. 12.25
2. 5.2
3. 10.75
4. 5.5

정답: 4

[4] TLB와 메모리의 Access time이 각각 20, 50ns인 경우 TLB의 사용이 유리하기 위한 Hit-ratio는 얼마 이상인가? p154 가상메모리

1. 0.5
2. 0.7
3. 0.6
4. 0.8

정답: (보기에 정답이 없어서 모두 정답처리 해주셨습니다.)

hit-ratio를 n 이라 했을 때 실주소로 메모리를 접근하는데 걸리는 시간 (실접근시간)은 $n * (\text{TLB접근} + \text{실주소접근}) + (1-n) * (\text{TLB접근} + \text{사상테이블접근} + \text{실주소접근})$

$$n * (20+50) + (1-n) * (20+50+50)$$

$$\text{TLB를 사용하지 않을 경우} = 50+50 = 100$$

$$70n + 120 * (1-n) = 120 - 50n < 100 \text{ 이어야 TLB의 사용이 유리하다.}$$

$$50n > 20$$

$$n > 0.4$$

[5] 다음 중 연관이 없는 것은?

1. Mutual exclusion
2. Page buffering p177 가상메모리관리(하드웨어) - 교체기법 pool의 사용
3. Semaphore
4. Critical section

[6] 다음 설명 중 틀린 것을 모두 고르시오. (하나 이상) p225 디스크 관리

1. Stripping과 Mirroring은 함께 사용 가능하다. RAID-1
 2. RAID에서 높은 level은 낮은 level들의 기능을 포함하지 않는다.
 3. Mirroring은 Disk system의 읽기/쓰기 속도를 높이는 방법이다. 복사본 만들어 신뢰도, 가용성을 높이는 법
 4. RAID 1의 신뢰도 항상 기법은 이보다 높은 level들의 방식과 다르다.
 5. Stripping은 신뢰도를 올리기 위한 방법이다. 저비용고성능 - 신뢰도 고려X
 6. Disk system의 Cylinder 수는 Disk 한 면의 Track 수와 같다. 실린더:같은 동심원의 트랙 기둥(논리 구조)
- 정답: 3,5

[7] 메모리에 추가의 프로세스를 받아들이는 기준을 Banker's algorithm에 기반할 경우, 현재 상황이 다음과 같다면 최초요구량 10, 최대요구량 25인 P4를 메모리에 추가할 수 있는가? (단, 메모리 전체량은 80임)

프로세스	현재사용량	최대요구량
1	10	30
2	20	50
3	30	40

p118 교착상태 회피기법

메모리 여유량 = $80 - (10+20+30) = 20$
P4를 받아들이면 메모리 여유량 10
10을 P3에 할당, 프로세스 완료 후 반납 -> 여유량 40
이후에 프로세스를 하나씩 완료하면 됨
>>> 안전상태

1. 받아들여도 된다.

2. 받아들이면 안 된다.

정답: 1

[8] Frame 4개를 부여받은 프로세스에 LRU-stack을 적용하여 Page replacement를 진행할 때, 다음과 같은 참조 열로 실행된 후의 stack 모양은? (각 답안의 오른쪽이 stack top이라 가정할 것)

참조 열(Reference string): a b a c b d b e d

1. b e c d p173 가상메모리관리 (하드웨어) 교체정책

2. c b e d

a b a c b d b e d

3. a c d e

4. d c b e

x x x x d b e d --- (top)

정답: c b e d

x x x c b d b e

x b a c c c d b

a a b b a a c c --- (bottom)

LRU스택은 bottom의 페이지가 교체 대상
참조되면 top으로 올린다

[9] 다음 중 다른 것과 deadlock 해결 방식이 다른 것은?

1. All or none Allocation p115 예방 - 자원의 부분 할당을 배제
2. Banker's algorithm p117 회피
3. Linear ordering p116 예방 - 자원의 환형 대기상황을 배제
4. Graph reduction p122 탐지

정답: 4 1, 2, 3은 deadlock 발생 자체를 막지만 4는 deadlock의 발생을 허용, 발생시 대처하는 방식

[10] 다음 중 디스크 입출력 요청이 충분히 있다는 가정 하에서 처리량과 분산(variance)이 가장 우수한 Disk scheduling 짝은?

1. FIFO-SSTF(Shortest seektime first) p212
2. SSTF-C-SCAN p218 c스캔은 한방향으로만 처리, 분산이 가장 우수함
3. SCAN-C-SCAN
4. SSTF-SCAN p214 처리량이 가장 우수한 SSTF + p215 처리량과 응답시간이 우수한 SCAN

정답: 4

[11] 다음 중 나머지 셋과 관계가 먼 것은?

1. Interrupt
2. Coalescing p141 메모리 관리 - 인접한 빈 공간의 병합
3. Polling
4. Trap

정답: 2

[12] 실행중인 프로세스는 여러 이유에 의해 다른 상태로 바뀔 수 있다.

다음 중 실행 상태(Running State)에서 바로 바뀔 수 있는 상태가 아닌 것은?

1. Ready state
2. Suspended ready state
3. Suspended blocked state
4. Blocked state

정답: 3

[13] 다음중 상호배제를 해결하는 접근방식이 다른 것은?

1. Peterson's algorithm
2. Test and Set instruction
3. Interrupt disable/enable
4. Exchange instruction

정답: 1

[14] 다음 중 주기억 장치에 적재된 후부터의 참조된 횟수 또는 시간에 대한 정보가 필요 없는 교체정책은?

1. FIFO (First in first out) p172
2. Second chance p174
3. LRU (Least recently used)p173
4. LFU (Least frequently used)p177

정답: 1

[15] 다음 중 선점형(Preemptive) 스케줄링 정책은?

1. Round Robin
2. FIFO (First in first out)
3. Shortest job first
4. Highest response-ration-next

정답: 1

[16] 다음 중 프로세스의 상호배제(Mutual exclusion) 목적으로 사용되지 않는 도구는?

1. Dekker's 알고리즘
2. 모니터 (Monitor)
3. Cycle stealing
4. Lamport의 Bakery algorithm

정답: 3

[17] 다음 중 틀린 설명을 모두 고르시오.

1. Multiprocessing은 Multiprogramming을 전제하여야 한다.
2. 처리량(throughput)과 응답시간(response time)은 상충되는 관계이다.
3. EDF(Earlist deadline first)는 RM(Rate monotonic)에 비해 스케줄 가능성을 높일 수 있다.
4. Interrupt의 중첩처리와 순차처리는 함께 사용할 수 없다.
5. User level thread는 운영체제와 독립적으로 운영될 수 있다.
6. 메모리 통합(Compaction)은 프로세스의 실행 도중 같이 실행될 수 있다.

정답: 4, 6

[18] 다음 중 입출력의 성능을 높이기 위한 방법이 아닌 것은?

1. 기억장치 인터리빙(storage interleaving)
2. 스푼링(Spooling)
3. Aging(Time stamping) 기법
4. DMA(Direct Memory Access)

정답: 3