

지난 시험에서 출제된 문제의 일부

(주의: 예상문제가 아님)

■ 다음 물음에 답하시오. 단, 서술형 답은 두(2) 줄 이내로 작성하시오. (각 문제당 5점)

1. 운영체제(OS)은 “control program”으로 정의될 수 있다. 이에 비추어 볼 때 운영체제의 목적이나 역할 중 가장 중요한 것은 무엇인가? (답: 사용자 프로그램의 실행)
3. 운영체제의 관점에서 인터럽트가 발생할 때, 어떤 일에 직접 개입하여야 할까? 가장 적합한 것을 하나 고르시오. ① 프로그램 카운터(program counter, PC)가 다음 명령을 가리키게 한다. ② DMA(direct memory access)인지 여부를 결정한다. ③ 인터럽트를 트랩(trap)으로 처리하게 한다. ④ 인터럽트 서비스 루틴, 즉 인터럽트가 발생하면 실행되는 프로그램을 준비해 놓아야 한다. ⑤ 인터럽트가 불능화(disable)되었는지 판단한다. (답: 4)
5. 대부분의 운영체제는 사용자 모드와 시스템 모드를 따로 운영하는 dual-mode operation을 채택하고 있다. 이렇게 하는 가장 중요한 이유는 무엇인가? (답: 시스템 레벨의 구성요소를 보호, or 자원공유의 정확성 유지, 등)
- 7-8. 디스크에 저장된 파일을 읽어오는 기능을 시스템 콜(system call)로서 커널 레벨에서 구현한다고 하자. (1) 일반적으로 시스템 콜이란 무엇을 의미하는가? 즉, 정의하시오. (2) dual mode operation을 지원하는 운영체제에서, 사용자가 시스템 콜을 호출하면 어떻게 처리되는가? (3) 이 시스템 콜에 대한 API를 예시로서 보이시오. (답: 1. 운영체제 서비스에 대한 프로그래밍 인터페이스 2. 커널이 시스템 레벨에서 처리 3. (예) 이름: Readfile; 인자: file handle, buffer address, #bytes, etc; **사용지침서**)
9. 사용자 인터페이스의 구현에 있어, 기능적으로 계층적인 구성을 갖는다고 하자. 다음의 기능들을 이용하여 사용자 인터페이스 모듈을 계층적으로 보이시오. {shell, GUI, command interpreter} (답: (맨위) GUI - shell - command interpreter (맨아래))
13. 운영체제에서 프로세스는 흔히 “program in execution”이라고 정의된다. 즉 프로그램이 실행되는 동적인 개체를 지칭한다. 이러한 동적인 상황을 표시하는 항목을 네 가지만 드시오. (답: {PCB에 들어가는 모든 항목, 프로세스의 상태, context switching, 생성/소멸, 등})
15. 오른쪽 코드에서 “wait()”는 어느 프로세스가 어느 프로세스를 언제까지 기다린다는 뜻인가? (답: 부모프로세스: 자식프로세스가 끝날 때까지 기다림)
17. 두 개의 프로세스가 데이터를 주고 받을 때, 송신측은 데이터를 송신하고 나서, 수신측이 데이터를 받았는지 확인하지 않고, 계속해서 다음 연산을 수행한다고 한다. 이러한 방법을 무엇이라 부르는지 해당되는 용어를 모두 고르시오. ① non-blocking send ② asynchronous send ③ blocking send ④ synchronous send ⑤ blocking receive (답: 1, 2)
19. 쓰레드(thread, lightweight process)는 한 개의 프로세스 내에서 정의될 수 있다. 이때 쓰레드가 (공유하지 않고) 독자적으로 갖는 정보(context, system data)는 무엇이 있는가? (답: 다음 다섯 개 중 세 가지 이상: 프로그램 카운터(PC), 범용 레지스터, 스택 포인터, condition code(status) register, 스택, thread id)
21. 어떤 운영체제에서 커널 레벨의 쓰레드를 제공하지 않는다고 한다. 이 경우 사용자 레

벨에서 쓰레드를 만들어 사용할 수 있다. 이때, 한 쓰레드가 수행되다가 입출력 작업이 필요하여, 대기(wait) 상태에 들어갔다고 한다. 그렇다면 동료 쓰레드(peer thread)가 그 대기상태 동안에 수행될 수 있는가? (예, 아니오) (답: 아니오)

23. CPU scheduler는 어느 컴퓨팅 자원을 배분하는가? (답: cpu time)

25. 실시간 스케줄링(real-time scheduling)을 실현하는데 있어, preemptive kernel과 nonpreemptive kernel 중 어느 것이 적합한가? 그리고 그 이유는? 단, preemptive kernel이란 어떤 프로그램 P의 실행 중, 우선순위가 높은 프로그램의 실행을 위해 P의 실행은 중단시킬 수 있는 커널을 일컫는다. (답: preemptive kernel, response time을 줄이기 위해서)

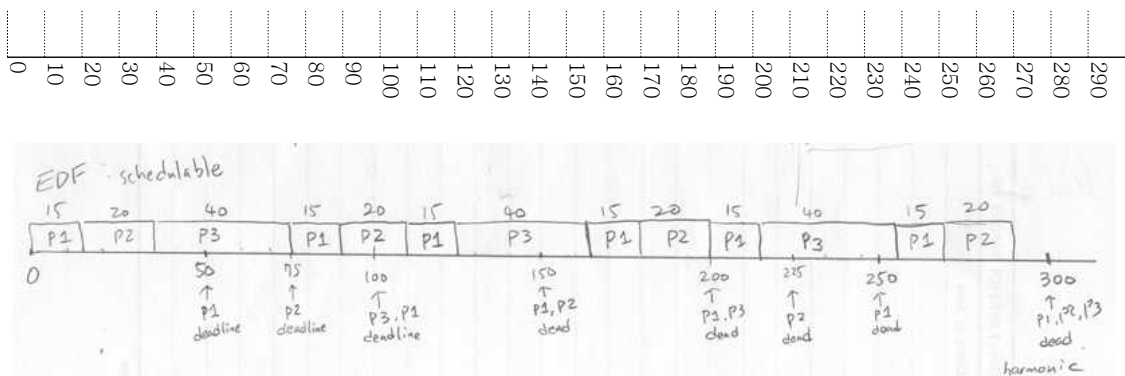
27. 다음의 용어를 모두 이용하여 프로세스 동기화(synchronization)가 무엇인지 간략히 3줄 이내로 설명하시오. 뜻이 달라지지 않는 범위에서 용어의 변형은 허용됨. concurrency or concurrent execution (병행성 혹은 병행수행), data consistency (데이터 일관성), mutual exclusion (상호배제), interleaving (인터리빙), critical section (임계영역) (답 예시: 여러 프로세스를 인터리빙하면서 병행수행할 때 임계영역에 대한 상호배제 원칙을 적용하여 데이터의 일관성을 유지하는 행위)

29. bounded buffer problem에서 두 개의 서로다른 특성을 갖는 세마포어를 사용한다. 즉 세마포어로서 mutex와 empty(or full)을 사용하는데, 이 둘의 차이점은 무엇인가? (답: mutex: 상호배제원칙 구현, empty: 접근 순서(즉 동기화) 구현)

■ Consider three processes, P1, P2, and P3 where $p_1 = 50$, $t_1 = 15$, $p_2 = 75$, and $t_2 = 20$, and $p_3 = 100$ and $t_3 = 40$. (p: period, t: execution time)

(b) Illustrate the scheduling of these three processes using earliest-deadline-first (EDF) scheduling. (15 pts)

(b)



■ The following program implements the readers-writers problem using semaphores, where readers have priority. To allow multiple readers, we require that, when there are no readers reading, the first reader that attempts to read should wait on a semaphore. When there is already at least one reader reading, subsequent readers need not wait before entering. (20 pts)

(a) Fill in the blank with appropriate code. (Ans: wait(wsem))

```
int readcount = 0;
semaphore x = 1, wsem = 1;
void reader()
{
    while (true)
    {
        wait(x);
        readcount++;
        if (readcount == 1)
            _____; /* (a) */
        signal(x);
        READ();
        wait(x);
        readcount--;
        if (readcount == 0) signal(wsem); /* (b) */
        signal(x);
    }
}

void writer()
{
    while (true)
    {
        wait(wsem);
        WRITE();
        signal(wsem);
    }
}
```

- Based on Banker's algorithm for deadlock avoidance, determine whether the state shown in the following table is safe or unsafe. Also, explain why or how. (20pts)

{Ans: safe. possible sequence <P1, P3, P0, P2> }

Processes	Current allocation		Maximum allocation		Resources available		Resources that may be needed	
	A	B	A	B	A	B	A	B
P0	1	2	4	2	1	1		
P1	0	1	1	2				
P2	1	0	1	3				
P4	2	0	3	2				