

# 4 차 종합 평가

작성자: 이상훈

본 자료는 모두 필자(이상훈, gcccompil3r@gmail.com)가 작성한 것이며  
해당 자료를 무단으로 불펌할 경우 발생하는  
불이익(형사 소송등등)에 대해 일절 책임지지 않습니다.

\* 모든 프로그래밍 문제는 함수를 작성하지 않을 경우 40% 감점

1. 그간 수업을 진행해오면서 느낀점을 3 줄 이상 기술하시오. (배점: 10 점)
2. 리눅스 커널은 운영체제(OS)다.  
OS가 관리해야 하는 제일 중요한 5 가지에 대해 기술하시오.
3. 프로세스와 스레드를 구별하는 방법에 대해 기술하시오.
4. Kernel 입장에서 Process 혹은 Thread를 만들면 무엇을 생성하는가 ?
5. 리눅스 커널 소스에 보면 struct task\_struct \*current 라는 것이 보인다  
이것이 무엇을 의미하는 것인지 적으시오.
6. 현재 4 개의 CPU(0, 1, 2, 3)가 있고 각각의 RQ 에는 1, 2 개의 프로세스가 위치한다.  
이 경우 2 번 CPU 에 있는 부모가 fork()를 수행하여 Task 를 만들어냈다.  
이 Task 는 어디에 위치하는 것이 좋을까 ?  
그리고 그 이유를 적으시오.
7. 위의 문제에서 이번에는 0, 1, 3 에 매우 많은 프로세스가 존재한다.  
이 경우 3 번에서 fork()를 수행하여 Task 를 만들었다.  
이 Task 는 어디에 위치하는 것이 좋을까 ?  
역시 이유를 적으시오.

8. UMA 와 NUMA 에 대해 기술하고

Kernel 에서 이들을 어떠한 방식으로 관리하는지 기술하시오.

9. Linux Kernel 은 외부 단편화와

메모리 부하를 최소화하기 위해 Buddy 할당자를 사용한다.

Buddy 할당자의 Algorithm 을 자세히 기술하시오.

10. 내부 단편화를 최소화 하기 위해

Buddy 에서 Page 를 받아 Slab 할당자를 사용한다.

Slab 할당자는 어떤식으로 관리되는지 기술하시오.

11. Task 구조체의 Pointer에 연결되어 있는 구조체들중

inode와 super\_block은 어떠한 정보를 포함하는지 기술하시오.

12. System Call 호출시 Kernel 에서 실제 System Call 을 처리하기 위해

Indexing 을 수행하여 적절한 함수가 호출되도록 주소값을 저장해놓고 있다.

이 구조체의 이름을 적으시오.

13. User Space 에서 System Call 번호를 전달한다.

Intel Machine 에서는 이를 어디에 저장하는가 ?

또한 ARM Machine 에서는 이를 어디에 저장하는가 ?

14. Micro Kernel 의 특징을 기술하시오.

15. Monolithic Kernel 의 특징을 기술하시오.

16. Linux Kernel 은 Micro 혹은 Monolithic 어떠한 것인가 ?

17. Linux 에 Device Driver 는 어떠한 형태인가 ?

그리고 이를 취함으로써 얻게된 이점은 무엇인가 ?

18. module\_init() 함수 호출은 언제 이루어지는가 ?

19. `module_exit()` 함수 호출은 언제 이루어지는가 ?
20. Device Driver 작성시 Wrapping 해야하는 부분이 어디인가 ?  
(Task 구조체에서 부터 연결된 부분까지를 꼭 이어서 작성하라)
21. Device Driver 는 Major Number 와 Minor Number 를 통해 Device 를 관리한다.  
실제 Device 의 Major Number 와 Minor Number 를 저장하는 변수는  
어떤 구조체의 어떤 변수인가 ?  
(역시 Task 구조체에서부터 꼭 찾아오길 바람)
22. Kernel 자체에 `kmalloc()`, `vmalloc()`, `__get_free_pages()`를 통해  
메모리를 할당할 수 있다.  
또한 `kfree()`, `vfree()`, `free_pages()`를 통해 할당한 메모리를 해제할 수 있다.  
이러한 Mechanism 이 필요한 이유가 무엇인지 자세히 기술하라.
23. Character Device Driver 를 아래와 같이 동작하게 만드시오.  
`read(fd, 1, 10)`을 동작시킬 경우  
2 번째 인자 ~ 3 번째 인자 까지의 덧셈을 반환하도록 한다.  
`write(fd, 1, 5)`를 동작시킬 경우  
2 번째 인자 ~ 5 번째 인자 까지의 곱셈을 반환하도록 한다.  
`lseek(fd, 7, SEEK_END)`를 동작시킬 경우 2 번째 인자값을 반환하고  
Kernel 내에서 "SEEK\_END means end"를 출력하게 하라!  
`close(fd)`를 수행하면 Kernel 내에서 "Finalize Device Driver"가 출력되게 하라!
24. CISC Architecture와 RISC Architecture에 대한 차이점을 기술하라.
25. Pipeline이 깨지는 경우에 대해 자세히 기술하시오.
26. 그동안 많은 것을 배웠을 것이다.  
최종적으로 수업시간에 그려줬던 Kernel Map 을 그려보도록 한다.  
(Networking 부분은 생략해도 좋다)  
예로는 다음을 생각해보도록 한다.  
게임을 더블 클릭하여 실행한다고 할 때  
그 과정 자체를 Linux Kernel 에 입각하여 기술하도록 하시오.  
(그림과 설명을 같이 넣어야함)
27. ARM 에서 인자 5 개를 사용하는 함수를 작성하고  
이에 대한 어셈블리어 분석을 수행하시오.  
동작 과정을 기술하기 위해 그림을 그리도록 한다.

28. Tree 의 삽입과 출력을 어셈블리어로 구현하시오.
29. 함수 포인터를 ARM 어셈블리어로 표현해보시오.
30. 첫 시험의 21 번 문제를 어셈블리어로 프로그래밍 하시오.
31. 파이프라인 구조에 대해 기술하시오.
32. 임의의 파일을 읽어서 내용을 출력하는  
리눅스 시스템 프로그램을 어셈블리어로 작성하시오.
33. TMS570 MCU 의 LED Blinky 를 순수한 어셈블리어로만 수행하시오.
34. 복소수 곱셈을 어셈블리어로작성하시오.