

1. 파이프 통신을 구현하고 c type.c 라고 입력할 경우
현재 위치의 디렉토리에 type.c 파일을 생성하도록 프로그래밍하시오.

2. 369 게임을 작성하시오.
2 초내에 값을 입력하게 하시오.
박수를 쳐야 하는 경우를 Ctrl + C 를 누르도록 한다.
2 초내에 값을 입력하지 못할 경우 게임이 오버되게 한다.
Ctrl + C 를 누르면 "Clap!"이라는 문자열이 출력되게 한다.

3. 리눅스 커널은 운영체제(OS)다.
OS 가 관리해야 하는 제일 중요한 5 가지에 대해 기술하시오.

~~~

파일시스템, 메모리매니지,태스크,네트워크,디바이스

물리적 cpu, 메모리, 디스크, 터미널, 네트워크  
추상적 태스크, 세그먼트/페이지, 파일, 프로토콜/패킷

~~~

4. Unix 계열의 모든 OS 는 모든 것을 무엇으로 관리하는가 ?

~~~~

태스크, 파일

~~~~

5. 리눅스에는 여러 장점이 있다.
아래의 장점들 각각에 대해 기술하라.

- * 사용자 임의대로 재구성이 가능하다.
- * 열악한 환경에서도 HW 자원을 적절히 활용하여 동작한다.
- * 커널의 크기가 작다.
- * 완벽한 멀티유저, 멀티태스킹 시스템
- * 뛰어난 안정성
- * 빠른 업그레이드
- * 강력한 네트워크 지원
- * 풍부한 소프트웨어

6. 32bit System 에서 User 와 Kernel 의 Space 구간을 적으시오.

32bit 에서 총 4GB 가상공간 할당중에

0~3GB 를 사용자 space 사용

3~4GB 를 커널 space 로 사용

7. Page Fault 가 발생했을때

운영체제가 어떻게 동작하는지 기술하시오.

free 한 페이지 프레임을 할당받고 해당 실행파일에서 필요한 페이지를 이페이지 프레임에 읽어다 놓게 됨
요구페이징

8. 리눅스 실행 파일 포맷이 무엇인지 적으시오.

ELF

9. 프로세스와 스레드를 구별하는 방법에 대해 기술하시오.

pid 가 생성되고 tgid 도 동시에 생성된다

프로세스는 메모리를 공유하지않기 때문에 tgid 값이 부모와 다르고

쓰레드는 메모리를 공유하기 때문에 tgid 값도 같다

tgid 값으로 구별할수있다

10. Kernel 입장에서 Process 혹은 Thread 를 만들면 무엇을 생성하는가 ?

~~~~~

task\_struct 를 생성

~~~~~

11. 리눅스 커널 소스에 보면 current 라는 것이 보인다.

이것이 무엇을 의미하는 것인지 적으시오.

커널 소스 코드와 함께 기술하시오.

~~~~~

현재 task\_struct 의 구조체를 가르킬수 있게 해줌

tgid 값을 리턴하기도함

~~~~~

12. Memory Management 입장에서 Process 와 Thread 의 핵심적인 차이점은 무엇인가 ?

13. Task 가 관리해야하는 3 가지 Context 가 있다.

System Context, Memory Context, HW Context 가 있다.

이중 HW Context 는 무엇을 하기 위한 구조인가 ?

~~~~~

실행중이던 태스크가 대기상태나 준비상태로 전이할 때 이 태스크가 어디까지 실행했는지 기억해두는 공간으로  
태스크가 다시실행될때 기억해 두었던 곳부터 다시 시작

~~~~~

14. 리눅스 커널의 스케줄링 정책중 Deadline 방식에 대해 기술하시오.

~~~~~

실시간 태스크 스케줄링 기법 중하나인 EDF 알고리즘을 구현한 것으로 비교적 최근에 추가되었고  
가장 가까운(급한) 태스크를 스케줄링 대상으로 선정하는 방식

~~~~~

15. TASK_INTERRUPTIBLE 과 TASK_UNINTERRUPTIBLE 은 왜 필요한지 기술하시오

~~~~~

대기하는동안 기다리는 그 사건외에 방해받으면 안되는 경우에는

TASK\_INTERRUPTIBLE

방해를 받아도 되는 경우에는

TASK\_UNINTERRUPTIBLE

이 것이 지켜지지않으면 먼저 해야되야 될것듯이 꼬여서 작업이 안됨

~~~~~

16. $O(N)$ 과 $O(1)$ Algorithm 에 대해 기술하시오.
그리고 무엇이 어떤 경우에 더 좋은지 기술하시오.

~~~~~

$O(N)$  : 단계의 수와 입력값이 1:1 관계를 가짐  
 $n$  이 커질수록 단계가 커짐  $O(1)$ 보다 느려짐  
1 이면 같음

$O(1)$  : 입력값  $n$  이 주어진다면 오직 한단계만 거침  
 $n$  이 커져도 오로지 한번만 함 ( 이 알고리즘이 가장 시간적으로 빠름 )

~~~~~

17. 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.

18. 15 번 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.

19. UMA 와 NUMA 에 대해 기술하고
Kernel 에서 이들을 어떠한 방식으로 관리하는지 기술하시오.
커널 내부의 소스 코드와 함께 기술하도록 하시오.

NUMA : CPU 들을 몇개의 그룹으로 나누고 각각의 그룹에게 별도의 지역메모리를 주는 구조가 생김
(CPU 에 메모리를 하나씩 병렬로 연결)

UMA : 모든 프로세스들이 상호간 연결되어 하나의 메모리를 공유 (메모리여러개가 버스에 연결)

20. Kernel 의 Scheduling Mechanism 에서 Static Priority 와 Dynamic Priority 번호가 어떻게 되는지 적으시오.

21. ZONE_HIGHMEM 에 대해 아는대로 기술하시오.

~~~~~

896MB 이상의 메모리 영역을 ZONE\_HIGHMEM 이라하며 이전의 큰메모리를 취급할때  
취약점에 대해서 보완하여 나온것

~~~~~

22. 물리 메모리의 최소 단위를 무엇이라고 하며 크기가 얼마인가 ?
그리고 이러한 개념을 SW 적으로 구현한 구조체의 이름은 무엇인가 ?

페이지프레임
4KB
page

23. Linux Kernel 은 외부 단편화와 메모리 부하를 최소화하기 위해 Buddy 할당자를 사용한다.
Buddy 할당자의 Algorithm 을 자세히 기술하시오.

zone 의 구조체에 존재하는 free_area[]배열에 구축되어 씌움

24. 21 번에 이어 내부 단편화를 최소화 하기 위해 Buddy 에서 Page 를 받아 Slab 할당자를 사용한다.
Slab 할당자는 어떤식으로 관리되는지 기술하시오.

공간을 미리분할해서 사용자가 요청하면 공간에서 떼어준다
그리고 해제 되면 다시 공간으로 가지고 있게됨 (버디에게 주는게아님)

25. Kernel 은 Memory Management 를 수행하기 위해 VM(가상 메모리)를 관리한다.
가상 메모리의 구조인 Stack, Heap, Data, Text 는 어디에 기록되는가 ?
(Task 구조체의 어떠한 구조체가 이를 표현하는지 기술하시오)

task_struct > mm_struct

26. 23 번에서 Stack, Heap, Data, Text 등 서로 같은 속성을 가진 Page 를 모아서 관리하기 위한 구조체 무엇인가?

(역시 Task 구조체의 어떠한 구조체에서 어떠한 식으로 연결되는지 기술하시오)

page

27. 프로그램을 실행한다고 하면 fork(), execve()의 콤보로 이어지게 된다.

이때 실제 gcc *.c 로 컴파일한 a.out 을 ./a.out 을 통해 실행한다고 가정한다.

실행을 한다고 하면 a.out File 의 Text 영역에 해당하는 부분을 읽어야 한다.

실제 Kernel 은 무엇을 읽고 이 영역들을 적절한 값으로 채워주는가?

28. User Space 에도 Stack 이 있고 Kernel Space 에도 Stack 이 존재한다.

좀 더 정확히는 각각에 모두 Stack, Heap, Data, Text 의 메모리 기본 구성이 존재한다.

그 이유에 대해 기술하시오.

29. VM(가상 메모리)와 PM(물리 메모리)를 관리하는데 있어

VM 을 PM 으로 변환시키는 Paging Mechanism 에 대해 Kernel 에 기반하여 서술하시오.

페이지 테이블을 통해 페이지징함

30. MMU(Memory Management Unit)의 주요 핵심 기능을 모두 적고 간략히 설명하시오.

가상주소로부터 물리주소로의 변환을 담당하는 별도의 하드웨어

페이지테이블의 시작점 pgd 와 변환할 가상주소를 입력하고 넣어주면됨

페이지테이블엔트리 캐시를 사용

31. 하드디스크의 최소 단위를 무엇이라 부르고 그 크기는 얼마인가?

섹터

기본적으로 512byte

32. Character 디바이스 드라이버를 작성할 때 반드시 Wrapping 해야 하는 부분이 어디인가?

(Task 구조체에서 부터 연결된 부분까지를 꼭 이어서 작성하라)

33. 예로 유저 영역에서 open 시스템 콜을 호출 했다고 가정할 때 커널에서는 어떤 함수가 동작하게 되는가?

실제 소스 코드 차원에서 이를 찾아서 기술하도록 한다.

34. task_struct 에서 super_block 이 하는 역할은 무엇인가 ?

file system 에 관한 전체적인 정보를 저장하는 block

35. VFS(Virtual File System)이 동작하는 Mechanism 에 대해 서술하시오.

b.txt 라는 파일을 open() 시스템 콜 하면

파일이 어떤파일시스템에 속해있는지 판단하고

파일의 정보를 담을 목적으로 여러가지 정보를 담기에 충분한 공간(구조체생성)

구조체를 인자로 하여 ext2 파일시스템 내부에 구현되는 open 함수를 호출

ext2 자기신의 inode 테이블을 뒤져서 b.txt 의 inode 를 찾아내고

inode 의 정보를 구조체에 넣은후 리턴

36. Linux Kernel 에서 Interrupt 를 크게 2 가지로 분류한다.

그 2 가지에 대해 각각 기술하고 간략히 설명하시오.

내부 인터럽트 : 현재 태스크와 관련이있는 동기적으로 발생하는 사건 “트랩” 이라고도 함

외부 인터럽트 : 현재 수행중인 태스크와 관련없는 주변장치에서 발생된 비동기 하드웨어사적인 사건

37. 내부 인터럽트는 다시 크게 3 분류로 나눌 수 있다.

3 가지를 분류하시오.

Fault, trap, abort

38. 35 번에서 분류한 녀석들의 특징에 대해 기술하시오.

39. 예로 모니터 3 개를 쓰는 경우 양쪽에 모두 인터럽트를 공유해야 한다.

Linux Kernel 에서는 어떠한 방법을 통해 이들을 공유하는가 ?

irq_desc 테이블 이용

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

Indexing 을 수행하여 적절한 함수가 호출되도록 주소값을 저장해놓고 있다.

이 구조체의 이름을 적으시오.

Inode

41. 38 에서 User Space 에서 System Call 번호를 전달한다.

Intel Machine 에서는 이를 어디에 저장하는가 ?

또한 ARM Machine 에서는 이를 어디에 저장하는가 ?

42. Paging Mechanism 에서 핵심이 되는 Page Directory 는 mm_struct 의 어떤 변수가 가지고 있는가 ?

43. 또한 Page Directory 를 가르키는 Intel 전용 Register 가 존재한다.

이 Register 의 이름을 적고 ARM 에서 이 역할을 하는 레지스터의 이름을 적으시오.

44. 커널 내부에서 메모리 할당이 필요한 이유는 무엇인가 ?

45. 메모리를 불연속적으로 할당하는 기법은 무엇인가 ?

46. 메모리를 연속적으로 할당하는 기법은 무엇인가 ?

47. Mutex 와 Semaphore 의 차이점을 기술하시오.

48. module_init() 함수 호출은 언제 이루어지는가 ?

49. module_exit() 함수 호출은 언제 이루어지는가 ?

50. thread_union 에 대해 기술하시오.

~~~~~`

태스크가 생성되면 리눅스는 task\_struct 구조체와 커널스택을 할당하고

태스크당 할당되는 커널스택을 thread\_union 이라한다

thread\_info 구조체를 포함

~~~~~


51. 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 쪽 찾아오길 바람)
52. 예로 간단한 Character Device Driver 를 작성했다고 가정해본다.
그리고 insmod 를 통해 Driver 를 Kernel 내에 삽입했으며
mknod 를 이용하여 /dev/장치파일을 생성하였다.
그리고 이에 적절한 User 프로그램을 동작시켰다.
이 Driver 가 실제 Kernel 에서 어떻게 관리되고 사용되는지 내부 Mechanism 을 기술하시오.
53. 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.
54. Character Device Driver 를 아래와 같이 동작하게 만드시오.
read(fd, buf, 10)을 동작시킬 경우 1 ~ 10 까지의 덧셈을 반환하도록 한다.
write(fd, buf, 5)를 동작시킬 경우 1 ~ 5 곱셈을 반환하도록 한다.
close(fd)를 수행하면 Kernel 내에서 "Finalize Device Driver"가 출력되게 하라!
55. OoO(Out-of-Order)인 비순차 실행에 대해 기술하라.
56. Compiler 의 Instruction Scheduling 에 대해 기술하라.

57. CISC Architecture 와 RISC Architecture 에 대한 차이점을 기술하라.

~~~~

CISC - 모든 가능한 경우를 고려해서 수백개의 명령어를 가지고 있다 .  
RISC - 자주 사용하는 명령어 약 40 개 정도만 사용하여 절약된 트랜지스터들로 성능을 향상 시킨다

~~~~

58. Compiler 의 Instruction Scheduling 은 Run-Time 이 아닌 Compile-Time 에 결정된다.
고로 이를 Static Instruction Scheduling 이라 할 수 있다.
Intel 계열의 Machine 에서는 Compiler 의 힘을 빌리지 않고도
어느저도의 Instruction Scheduling 을 HW 의 힘만으로 수행할 수 있다.
이러한 것을 무엇이라 부르는가 ?

59. Pipeline 이 깨지는 경우에 대해 자세히 기술하시오.

60. CPU 들은 각각 저마다 이것을 가지고 있다.

Compiler 개발자들은 이것을 고려해서 Compiler 를 만들어야 한다.

또한 HW 입장에서 이것을 고려해서 설계를 해야 한다.

여기서 말하는 이것이란 무엇인가 ?

~~~~~

버스인터페이스 유닛

~~~~~

61. Intel 의 Hyper Threading 기술에 대해 상세히 기술하시오.

~~~~~

2 개의 가상 프로세서를 구현하는 기술로 cpu 는 하나지만 운영체제는 2 개있는거처럼 두개의  
쓰레드를 처리할수있는 기술

~~~~~

62. 그동안 많은 것을 배웠을 것이다.

최종적으로 Kernel Map 을 그려보도록 한다.

(Networking 부분은 생략해도 좋다)

예로는 다음을 생각해보도록 한다.

여러분이 좋아하는 게임을 더블 클릭하여 실행한다고 할 때

그 과정 자체를 Linux Kernel 에 입각하여 기술하도록 하시오.

(그림과 설명을 같이 넣어서 해석하도록 한다)

소스 코드도 함께 추가하여 설명해야 한다.

63. 파일의 종류를 확인하는 프로그램을 작성하시도록 하시오.

64. 서버와 클라이언트가 1 초 마다 Hi, Hello 를 주고 받게 만드시오.

65. Shared Memory 를 통해 임의의 파일을 읽고 그 내용을 공유하도록 프로그래밍하시오.

66. 자신이 사용하는 리눅스 커널의 버전을 확인해보고

[https://mirrors.edge.kernel.org/pub/linux/kernel/v4.x/\\${자신의 버전}.tar.gz](https://mirrors.edge.kernel.org/pub/linux/kernel/v4.x/${자신의 버전}.tar.gz) 를 다운받아보고
이 압축된 파일을 압축 해제하고 task_struct 를 찾아보도록 한다.
일련의 과정을 기술하면 된다.

67. Multi-Tasking 의 원리에 대해 서술하시오.

(Run Queue, Wait Queue, CPU 에 기초하여 서술하시오)

68. 현재 삽입된 디바이스 드라이버의 리스트를 보는 명령어는 무엇인가 ?
69. System Call Mechanism 에 대해 기술하시오.
70. Process 와 VM 과의 관계에 대해 기술하시오.
71. 인자로 파일을 입력 받아 해당 파일의 앞 부분 5 줄을 출력하고
추가적으로 뒷 부분의 5 줄을 출력하는 프로그램을 작성하시오.
72. 디렉토리 내에 들어 있는 모든 File 들을 출력하는 Program 을 작성하시오.
73. Linux 에서 fork()를 수행하면 Process 를 생성한다.
이때 부모 프로세스를 gdb 에서 디버깅하고자하면 어떤 명령어를 입력해야 하는가 ?
74. C.O.W Architecture 에 대해 기술하시오.
75. Blocking 연산과 Non-Blocking 연산의 차이점에 대해 기술하시오.
76. 자식이 정상 종료되었는지 비정상 종료되었는지 파악하는 프로그램을 작성하시오.
77. 데몬 프로세스를 작성하시오.
잠시 동안 데몬이 아니고 영구히 데몬이 되게 하시오.
78. SIGINT 는 무시하고 SIGQUIT 을 맞으면 죽는 프로그램을 작성하시오.
79. goto 는 굉장히 유용한 C 언어 문법이다.
그러나 어떤 경우에는 goto 를 쓰기가 힘든 경우가 존재한다.
이 경우가 언제인지 기술하고 해당하는 경우
문제를 어떤식으로 해결 해야 하는지 프로그래밍 해보시오.

80. 리눅스에서 말하는 File Descriptor(fd)란 무엇인가 ?

~~~~~

시스템으로부터 할당 받은 파일 또는 소켓에 부여된 정수를 의미  
파일과 소켓은 생성과정에서 fd 할당

0:표준입력  
1:표준출력  
2:표준에러

~~~~~

81. `stat(argv[2], &buf)`일때 `stat` System Call 을 통해 채운 `buf.st_mode` 의 값에 대해 기술하시오.
82. 프로세스들은 최소 메모리를 관리하기 위한 `mm`, 파일 디스크립터인 `fd_array`, 그리고 `signal` 을 포함하고 있는데 그 이유에 대해 기술하시오.
83. 디렉토리를 만드는 명령어는 `mkdir` 명령어다.
`man -s2 mkdir` 을 활용하여 `mkdir` System Call 을 볼 수 있다.
이를 참고하여 디렉토리를 만드는 프로그램을 작성해보자!
84. 이번에는 랜덤한 이름(길어도 랜덤)을 가지도록 디렉토리를 3 개 만들어보자!
(너무 길면 힘드니까 적당한 크기로 잡도록함)
85. 랜덤한 이름을 가지도록 디렉토리 3 개를 만들고
각각의 디렉토리에 5 ~ 10 개 사이의 랜덤한 이름(길어도 랜덤)을 가지도록 파일을 만들어보자!
(너무 길면 힘드니까 적당한 크기로 잡도록함)
86. 85 번까지 진행된 상태에서 모든 디렉토리를 순회하며
3 개의 디렉토리와 그 안의 모든 파일들의 이름 중 `a`, `b`, `c` 가 1 개라도 들어있다면 이들을 출력하라!
출력할 때 디렉토리인지 파일인지 여부를 판별하도록 하시오.
87. 클라우드 기술의 핵심인 OS 가상화 기술에 대한 질문이다.
OS 가상화에서 핵심에 해당하는 3 가지를 기술하시오.
88. 반 인원이 모두 참여할 수 있는 채팅 프로그램을 구현하시오.
89. 88 번 답에 도배를 방지 기능을 추가하시오.
90. 89 번조차도 공격할 수 있는 프로그램을 작성하시오.
91. 네트워크 상에서 구조체를 전달할 수 있게 프로그래밍 하시오.
92. 91 번을 응용하여 `Queue` 와 `Network` 프로그래밍을 연동하시오.
93. `Critical Section` 이 무엇인지 기술하시오.

~~~~

일정영역의 코드가 실행되는 동안 다른 인터럽트가 걸리거나 다른일을 OS 가 스케줄을 바꿔거나 다른코드가 돌게 하는 일이없도록 하는 영역  
~~~~`

94. 유저에서 fork() 를 수행할때 벌어지는 일들 전부를
실제 소스 코드 차원에서 해석하도록 하시오.

95. 리눅스 커널의 arch 디렉토리에 대해서 설명하시오.

Alpha

arm

arm64

mips

ia64

sparc

x86

... 등 이있고

하드웨어 종속적인 부분들이 구현된 디렉터리임

보통 cpu 타입에 따라 하위 디렉토리가 다시 구분됨

96. 95 번 문제에서 arm 디렉토리 내부에 대해 설명하도록 하시오.

97. 리눅스 커널 arch 디렉토리에서 c6x 가 무엇인지 기술하시오.

TI DSP

98. Intel 아키텍처에서 실제 HW 인터럽트를
어떤 함수를 가지고 처리하게 되는지 코드와 함께 설명하시오.

99. ARM 에서 System Call 을 사용할 때 사용하는 레지스터를 적으시오.

유저 레지스터

100. 벌써 2 개월째에 접어들었다.

그동안 굉장히 많은 것들을 배웠을 것이다.

상당수가 새벽 3 ~ 5 시에 자서 2 ~ 4 시간 자면서 다녔다.

또한 수업 이후 저녁 시간에 남아서 9 시 ~ 10 시까지 공부를 한 사람들도 있다.

하루 하루에 대한 자기 자신의 반성과 그 날 해야할 일을 미루지는 않았는지 성찰할 필요가 있다.

그 날 해야할 일들이 쌓이고 쌓여서 결국에는 수습하지 못할정도로 많은 양이 쌓였을 수도 있다.

사람이란 것이 서 있으면 앉고 싶고 앉으면 눕고 싶고 누우면 자고 싶고 자면 일어나기 싫은 법이다.

내가 정말 죽을등 살등 이것을 이해하기 위해 열심히 했는지 고찰해보자!

2 개월간 자기 자신에 대한 반성과 성찰을 수행해보도록 한다.

또한 앞으로는 어떠한 자세로 임할 것인지 작성하시오.

첫달에는 남아서 많은 힘들어도 따라가려 많이 노력하였고, 처음이라 많은 시간을 할애하며 노력했었는데
하지만 시간이 지나며 조금씩 느슨해진거같은 생각이 들었고,

역시나 오늘 시험을 보면서 이번한달이 노력이 정말 부족하다는 것을 알수있게 되었다.

사실 첫달보다 복습을 많이 하지못했다 그래서 위에 쌓이고 쌓여서 많은 양이 쌓인거같은 생각이 든다.

문제를 풀고 그날의 수업을 이해하며 따라가려 하는 사람들을 보며 나는 그러지 않은거에 대해서 나도 같이 했어야 했는데 라며부끄러운 생각을 하게 되는거같다.

또한 지각을 많이 하는데 이것 또한 나의 마음가짐의 문제인거 같아서 앞으로는 그러지않기 위해 노력할것이고
아무래도 아무런 자극이없다면 3 달째는 더 풀어지지 않을까 싶었지만 오늘시험에 충격을 받고 더이상 마음가짐
이 흐트러지지 않게 할수있을 것 같다

독한 마음을 먹고 노력해야겠다.