

TI DSP, MCU 및 Xilinx Zynq FPGA

프로그래밍 전문가 과정

강사 - Innova Lee(이상훈)
gcccompil3r@gmail.com
학생 - 하성용
accept0108@naver.com

2 회차 시험

- 임베디드 애플리케이션 구현

03. 리눅스 커널은 운영체제(OS)다.

OS가 관리해야 하는 제일 중요한 5 가지에 대해 기술하시오.

CPU, 메모리, 디스크, 터미널, 네트워크

04. Unix 계열의 모든 OS는 모든 것을 무엇으로 관리하는가 ?

태스크 관리자, 메모리 관리자, 파일 시스템, 네트워크 관리자,
디바이스 드라이버 관리자

05. 리눅스의 장점에 대한 각각의 상세한 기술

리눅스에는 여러 장점이 있다. (배점 0.2 점)

아래의 장점들 각각에 대해 기술하라.

- * 사용자 임의대로 재구성이 가능하다.
 - 실력이 출중한 사용자 임의대로 재구성이 가능하다.
- * 열악한 환경에서도 HW 자원을 적절히 활용하여 동작한다.
 - 오래된 컴퓨터에서도 리눅스 설치가능
- * 커널의 크기가 작다.
 - 윈도우 커널보다 작음, 사용자 입장에서 방대함
- * 완벽한 멀티유저, 멀티태스킹 시스템
- * 뛰어난 안정성
 - ex) 구글
- * 빠른 업그레이드
 - 사용자 입장에서는 단점(자주 업데이트)
- * 강력한 네트워크 지원
- * 풍부한 소프트웨어
 - 스톨만 GNU

06. 32 bit System에서 User와 Kernel의 Space 구간을 적으시오.

3G

07. Page Fault가 발생했을때 운영체제가 어떻게 동작하는지 기술하시오.

page fault가 발생하면 메모리가 없으니까 page를 할당해줌
page handler가 동작해서 user에서 구동되었는지 kernel에서
구동되었는지 확인하고 kernel에서 구동된거면 page를 할당해주고
user에서 접근한거면 권한이 없다고 segmentation fault 나옴

08. 리눅스 실행 파일 포맷이 무엇인지 적으시오.

ELF

09. 프로세스와 스레드를 구별하는 방법에 대해 기술하시오.

프로세스 생성 함수 `fork()` 사용
스레드 생성 함수 `clone()` 사용

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

`task_struct` 라는 동일한 자료구조를 생성

11. 리눅스 커널 소스에 보면 `current` 라는 것이 보인다.
이것이 무엇을 의미하는 것인지 적으시오.

커널 소스 코드와 함께 기술하시오.

`current` 라는 매크로는 커널 내부에 정의되어 있는 매크로로써 현재 태스크의 `task_struct` 구조체를 가리킬 수 있게 해준다

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 ?

실행 중이던 태스크가 대기 상태나 준비 상태로 전이할 때 이 태스크가 어디까지 실행했는지 기억해 두는 공간으로, 이후 이 태스크가 다시 실행될 때 기억해 두었던 곳부터 다시 시작하게된다.

14. 리눅스 커널의 스케줄링 정책중 Deadline 방식에 대해 기술하시오.

`deadline` 이 가장 가까운(즉, 가장 급한) 마감시간이 가까운 태스크를 스케줄링 대상으로 선정한다

15. `TASK_INTERRUPTIBLE` 과 `TASK_UNINTERRUPTIBLE` 은 왜 필요한지 기술하시오.

`TASK_RUNNING` 상태에 있던 태스크가 특정한 사건을 기다려야 할 필요가 있으면 대기상태로 전이해야하는데 필요하다(ex.전화)
`TASK_INTERRUPTIBLE` 과 `TASK_UNINTERRUPTIBLE` 상태 모두 특정 사건을 기다린다는 면에서 유사하나 `TASK_UNINTERRUPTIBLE` 상태는 시그널에 반응하지않는다는 점에서 `TASK_INTERRUPTIBLE` 과 구분된다

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

처리해야할 양이 적으면 $O(n)$ 이 더 빨라서 좋고
처리해야할 양이 많으면 $O(1)$ 이 더 빨라서 좋다.

19. UMA 와 NUMA 에 대해 기술하고 Kernel 에서 이들을 어떠한 방식으로 관리하는지 기술하시오.

커널 내부의 소스 코드와 함께 기술하도록 하시오.

복수개의 CPU 가 메모리 등의 자원을 공유하기 때문에 성능상 병목현상이 발생하는데 따라서 CPU 들을 몇개의 그룹으로 나누고 각각의 그

룹에게 별도의 지역메모리를 주는 구조를 NUMA 라고 한다.
이에 반해 기존 시스템은 UMA 라고한다

21. ZONE_HIGHMEM 에 대해 아는대로 기술하시오.

상위 메모리

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

바이트라고 하며 크기는 4KB(4096 byte)이다

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

버디 할당자는 zone 구조체에 존재하는 free_area[]배열을 통해
구축된다.
free_area 배열은 10 개의 엔트리를 가진다
0~9 까지 각각의 숫자는 해당 free_area가 관리하는 할당의 크기를
나타낸다.
예를 들어 0 인경우 2 의 0 승 즉, 1 개의 페이지프레임이 할당의 단위
임을 뜻하고 1 인경우 2 의 1 승 즉, 2 개의 페이지 프레임이 할당의
단위임을 뜻한다
리눅스 구현상 최대 할당 크기는 4MB 이다(2n 승*4KB)

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

slab 할당자는 미리 페이지프레임의 크기만큼 할당받은 뒤 이 공간을
64byte 크기로 분할해둔다. 사용자가 64byte 의 공간을 요청하면
버디 할당자로부터 할당받아오는 것이 아니라 미리 할당받아 분할하여
관리하고 있던 바로 이 공간에서 떼어 주는 것이다. 추후 사용자가
할당받았던 64byte 의 공간을 해제한다면 역시 버디로 반납하는 것이
아니라 미리 할당 받아 관리하던 공간에서 다시 가지고 있으면 된다.
일종의 캐시로 사용하는 것이다.

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

물리메모리 매핑,
하드웨어 적인 리소스 사용을 가능하게해줌

31. 하드디스크의 최소 단위를 무엇이라 부르고 그 크기는 얼마인가 ?
2 의 12 승 byte = 4kb

45. 메모리를 불연속적으로 할당하는 기법은 무엇인가 ?

블록체인 기법, 인덱스 블록 기법

46. 메모리를 연속적으로 할당하는 기법은 무엇인가 ?

Sequential 할당 기법

50. thread_union 에 대해 기술하시오.

태스크당 할당되는 커널 스택

59. Pipeline 이 깨지는 경우에 대해 자세히 기술하시오.
call 이나 jmp 를 CPU Instruction(명령어) 레벨에서
분기 명령어라고 하고 이들은
CPU 파이프라인에 매우 치명적인 손실을 가져다준다.

74. C.O.W Architecture 에 대해 기술하시오.

Copy on write
쓰는 동시에 복사가 된다

75. Blocking 연산과 Non-Blocking 연산의 차이점에 대해 기술하시오.

Blocking : 값이 들어올 때까지 기다린다
Non-blocking : 읽을게 없으면 넘기고 읽을게 있으면 봄

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

93. Critical Section 이 무엇인지 기술하시오.

mutex 는 여러개의 스레드가 공유하는 데이터를 보호하기 위해서 사용되는 도구로써, 보호하고자 하는 데이터를 다루는 코드영역을 단지 한번에 하나의 스레드만 실행가능 하도록 하는 방법으로 공유되는 데이터를 보호한다. 이러한 코드영역(하나의 스레드만 점유가능한)을 critical section

96. 앞선 문제에서 arm 디렉토리 내부에 대해 설명하도록 하시오.

임베디드 디바이스용 디렉토리

97. 리눅스 커널 arch 디렉토리에서 c6x 가 무엇인지 기술하시오.
ti 사의 dsp

100. 20 점

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 죽을듯 살듯 이것을 이해하기 위해 열심히 했는지 고찰해보자!

2 개월간 자기 자신에 대한 반성과 성찰을 수행해보도록 한다.

(맨 처음 과정에 들어왔을 때의 모습과 현재의 모습에 어떠한 발전이 있었는지 등등을 자세히 기술하도록 한다)

또한 앞으로는 어떠한 자세로 임할 것인지 작성하시오.

나는 피곤하다는 핑계로 그날 해야할일들을 미루고 미루고 주말까지 미뤘다

그리고 주말에는 기다렸다는듯이 번거로운일이 생겨 수습을 하고나면 6 시간정도가 남았다

쿨근하고 퇴근하는 찰나의 시간을 이용해 영어라는 타국가의 언어를 마스터하는 사람들을 많이 보았다

그 사람들과 나의 차이는 그 1 분을 어떻게든 써먹느냐 그저 찰나로 취급하냐일 것이다

시간이 없었다 는 건 분명 핑계이다

같이 공부하는사람들은 주말 2 일이 있었지만 나는 6 시간밖에 없었어라고 어쩔 수 없었다고 한적도 있다

그리고 깨달은건 나는 시간이 있었다는것이다 어찌면 다른사람보다 더 많았을수도있다

프리한시간이 주말에는 6 시간이나 있었으며 평일에는 3 시간씩 남았다

합하면 21 시간이라는 거의 하루에 근접하는 시간이 있었다

아무 발전없이 그저 깨닫기만한건 적었고 어떠한 자세로 임할것인지를 적을 차례인데

솔직히 이건 쓰기가 두렵다

나는 약속이란걸 싫어하는사람이다

정확히는 지키지못할수도 있는 약속을 하는걸 싫어한다

나는 분명 나를 아는사람들에게 신뢰를 받지만 그 신뢰는 실력에 대한 신뢰가 아닌 바람을 맞히거나 뒤통수를 치지않는다는 인성에 대한 신뢰이다

저번 반성문에 아마 철저한 복습을 적었었는데 모호한 표현은 목표설정에 안좋다고 들었다

그래서 이번에는 조금이나마 확실한 현실적인 목표를 적으려한다

수업진도 이미 못따라갔다 과정이 끝날때까지 못따라갈수도 있다

그래서 내가 현재 현실적으로 할수있는걸 하려고 한다

1 일차부터 차근차근 복습해나갈 예정이다

중간부터 시작하려고도 했지만 아니었다 아예 처음부터 시작하는게 지금처럼 코
딩한줄한줄마다 의문을 가지는것보다 빠를것이라 확신한다

타이핑실력만 느는나날은 이제 그만하고싶다

다음 반성문때는 최선을 다했다 라는 한줄만 추가할수있으면 정말 좋을거같다