

' INTERRUPT & SYSTEM CALL ' IN LINUX KERNEL

Operating System & Computer Architecture/Linux

2017. 12. 28. 21:01

여기에 항상 할당되어 있음.

인터럽트발생 -> idt_table을 참조하여 해당인터럽트 실행

1.시스템콜 -> 시스템콜 테이블참조하여 시스템함수실행

2.인터럽트번호 32~255-> irq_desc참조하여 그 함수실행

IRQ에 인터럽트 33을 등록함.

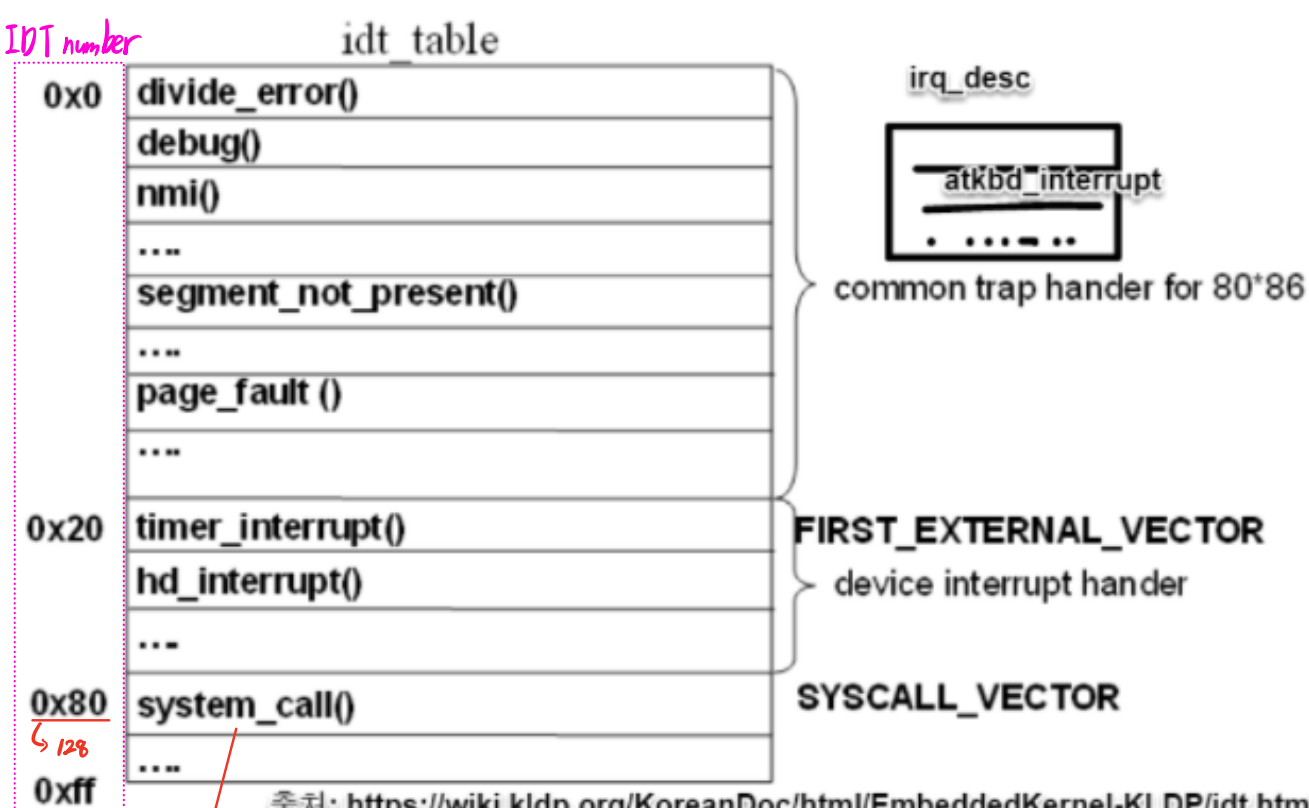
ex) 키보드입력

키보드 눌림 -> 인터럽트33번 cpu에게전달 -> cpu는처음 운영체제가 시작할때 메모리에 할당해준 IDT를 참조 -> 33번이기때문에

irq_desc로감 -> 33번에해당하는 irq_desc[1]을 부름-> irq_desc[1] 는 atkbd함수를 호출 -> atkbd함수실행

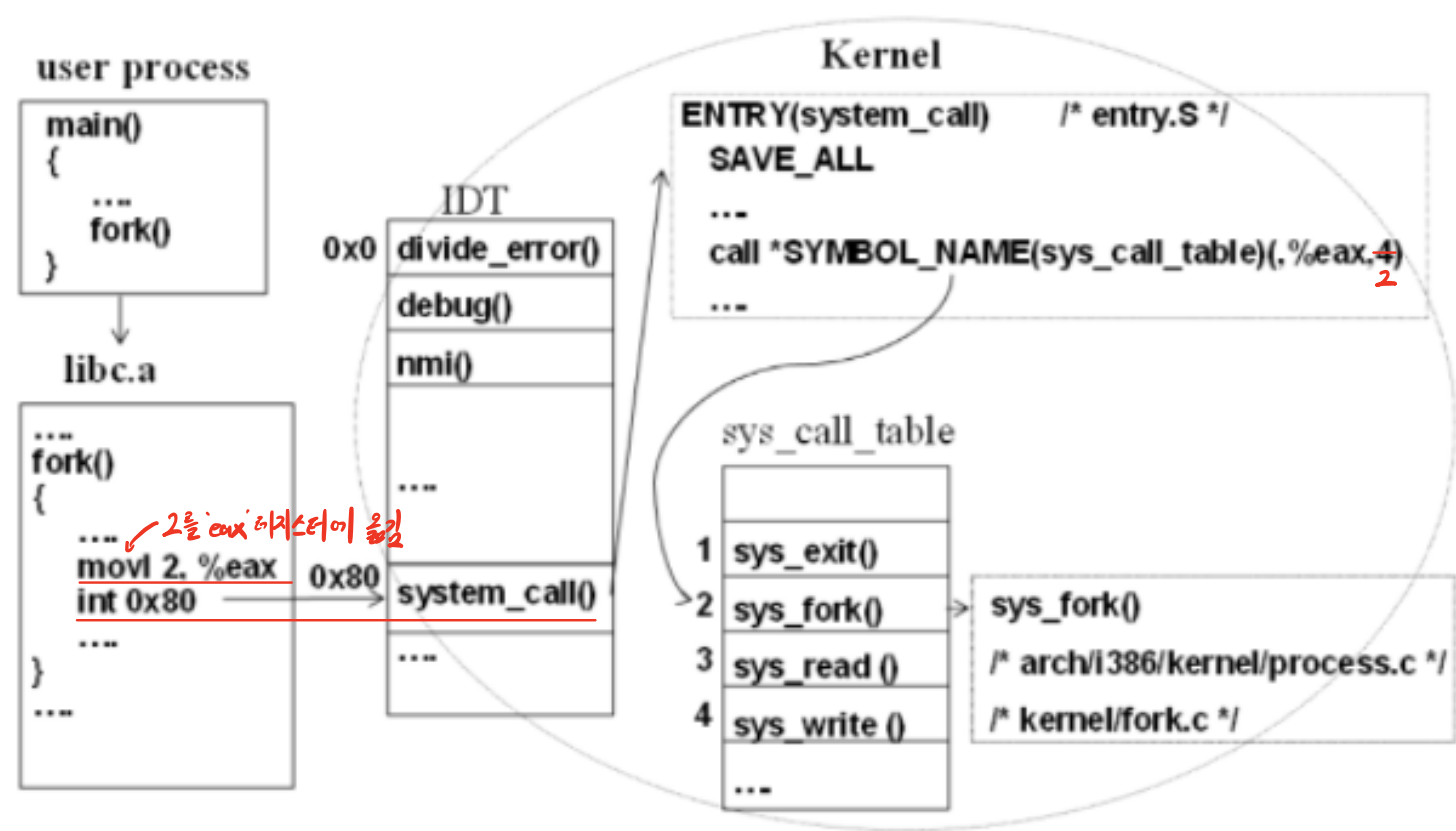
128번은 system_call 인터럽트 이고 타이머인터럽트,페이지폴트,키보드입력(그림에는없지만 아마 33)은 다른번호에 배정받아있다.

IDT number



System_call일때는 IDT넘버가 128번이고 sys_call_table을 참조하여 그에해당하는 커널함수를 알수있다.

그림 B-2. fork()가 실행될 때의 흐름



fork가 실행될때 간단설명

- 1. `mov eax = 2` //시스템콜번호를 레지스터에저장
INT 128
- 2. IDT[128] //idt테이블에서 128번으로가라 = system_call()
- 3. sys_call_table[4] //sys_call_tale 4번으로가서 실행해라.
2