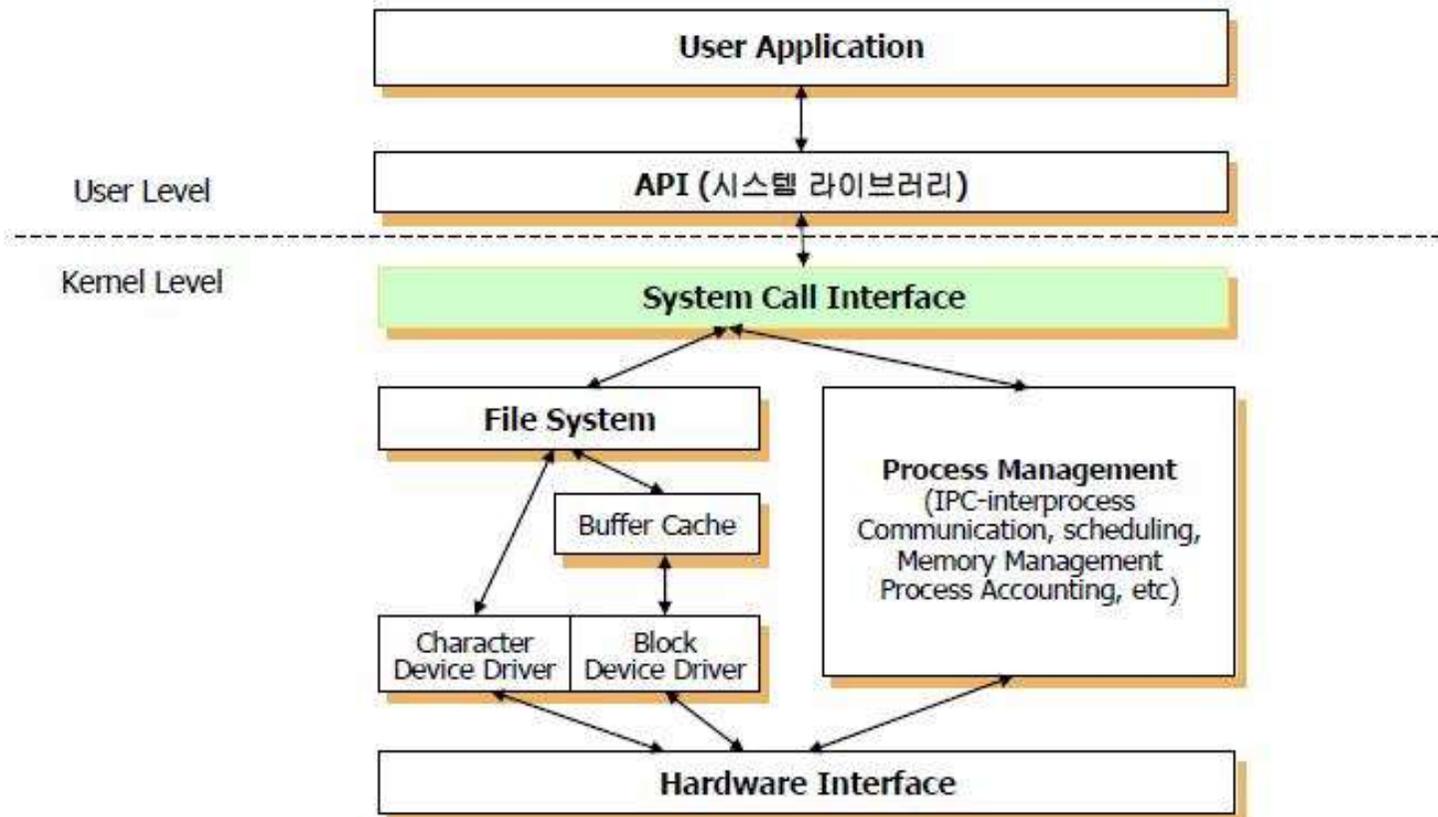


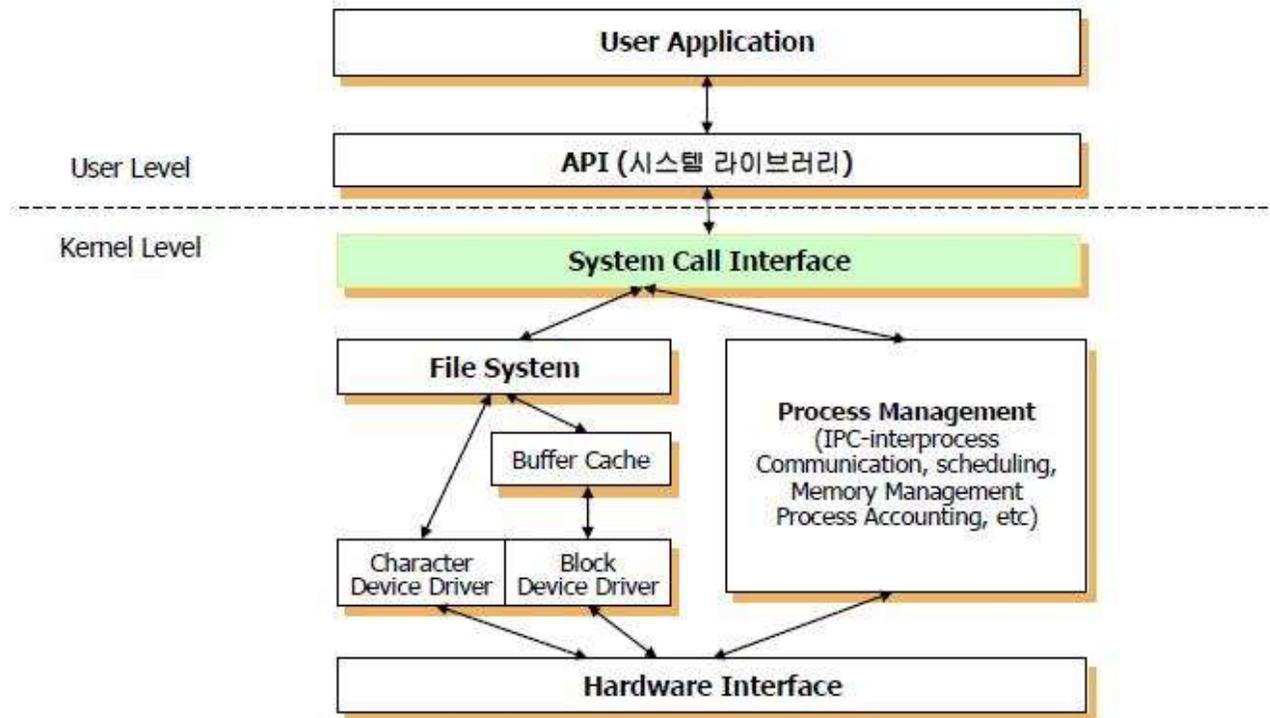
System Call

System Call



System Call

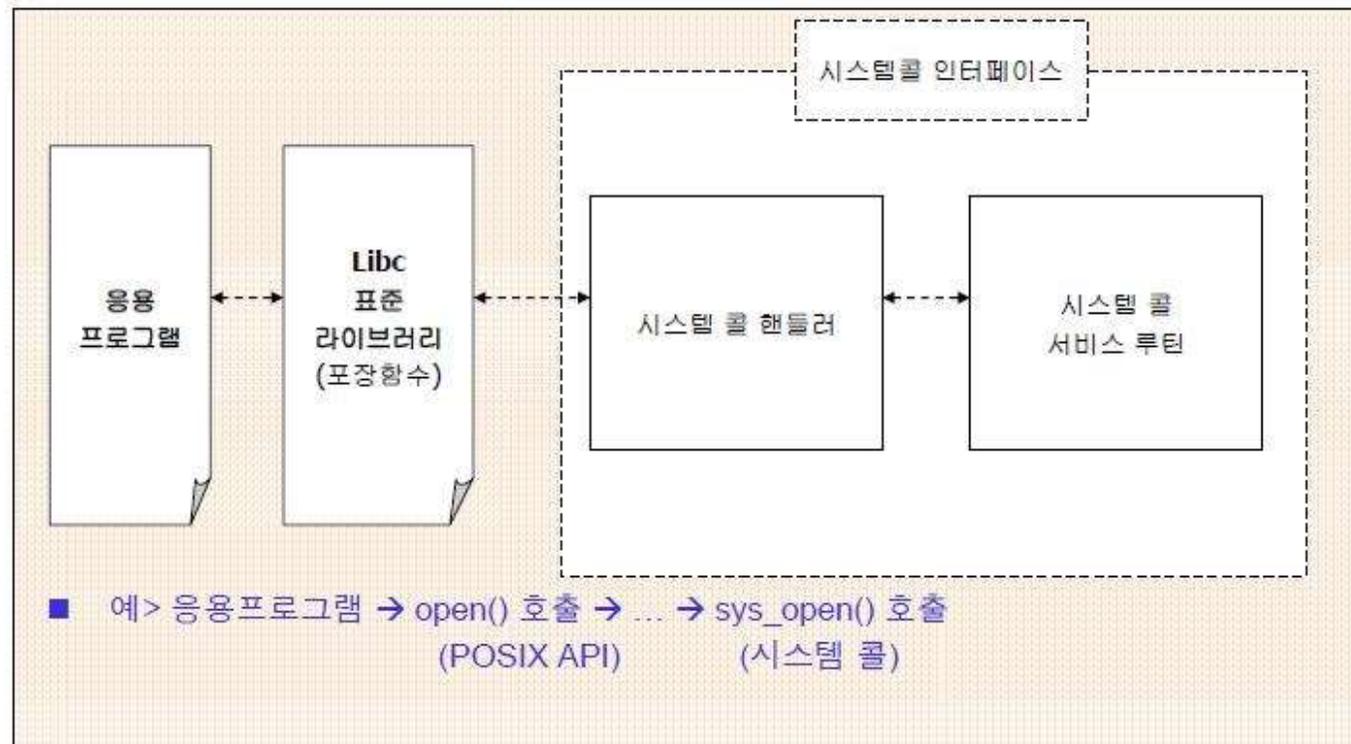
- User mode process 와 kernel 간의 interface
 - ✓ User mode process는 kernel 영역에 접근할 수 없다
 - Kernel의 자료구조 및 HW에 대한 접근 불가
 - ✓ User mode process가 kernel이 가지고 있는 시스템 상태정보를 열람하거나 HW에 접근하여 제어하기 위해 kernel과의 통신이 필요



POSIX API & System Calls

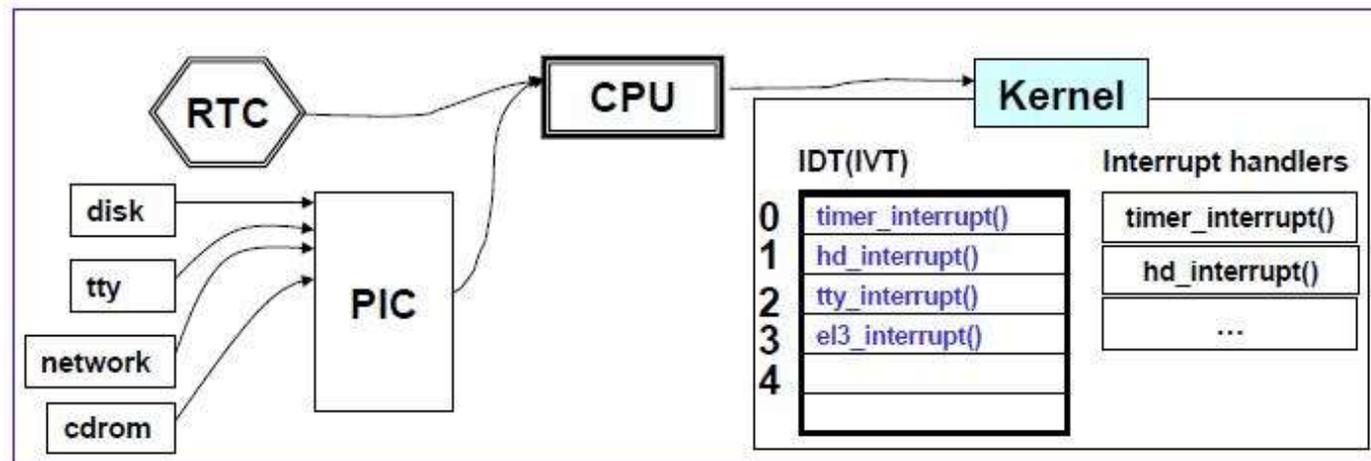
- POSIX API (Application Programming Interface)
 - ✓ 유닉스 기반의 표준 운영체제 인터페이스
 - ✓ Application이 시스템에 서비스를 요청할 때 어떤 함수를 사용해야 하는 지 지정
 - ✓ 표준을 두어 다른 시스템에 응용 프로그램을 porting 하는 것이 용이하게 하기 위한 목적
 - ✓ Open(), close(), read(), write() 등
- System Call
 - ✓ SW인터럽트를 통해 커널에 서비스를 요청
 - ✓ Linux에서는 POSIX API를 준수하는 library 함수 안에서 system call 함수를 호출

POSIX API & System Calls

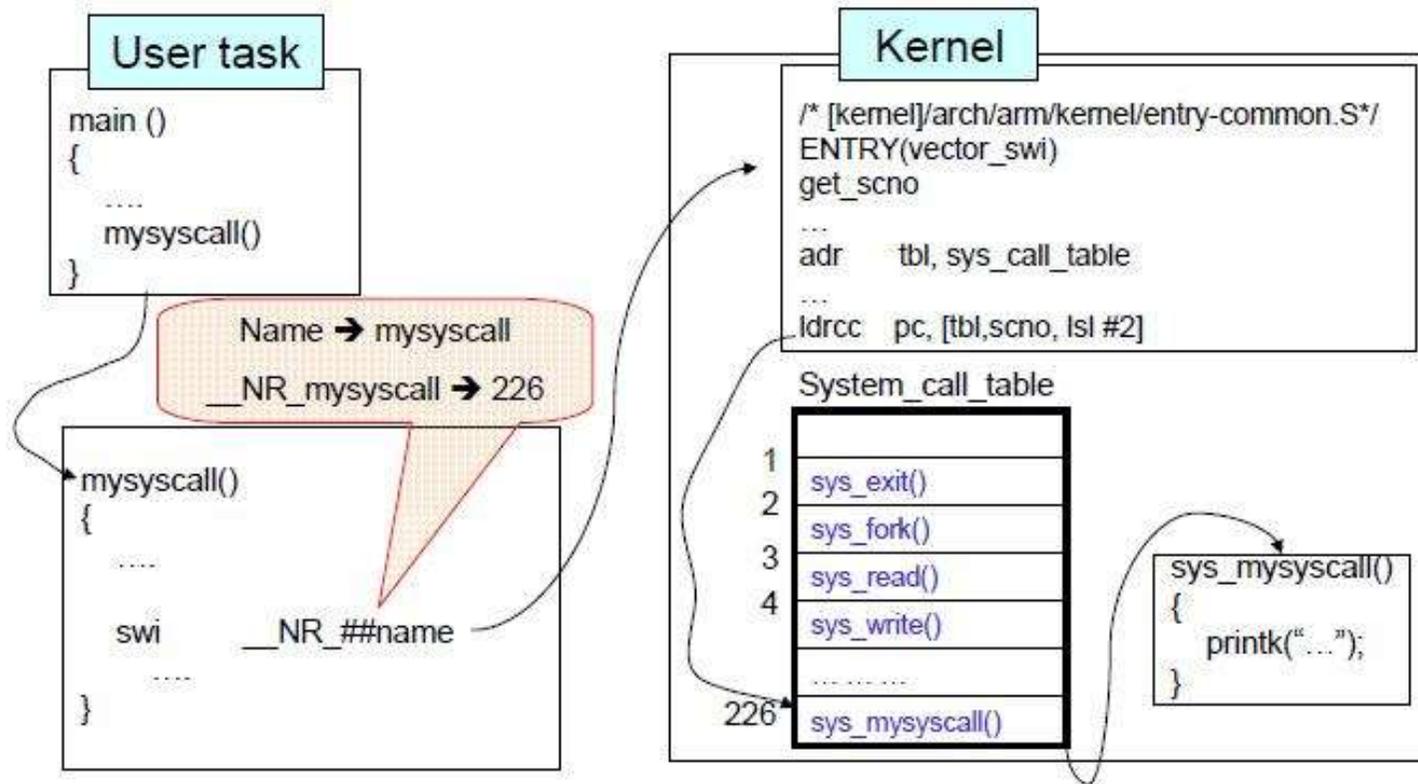


System Call

- Linux에서의 system call 처리
 - ✓ Interrupt 처리 메커니즘 사용
- Interrupt
 - ✓ 주변 장치와 커널이 통신하는 방식 중 하나로 주변 장치가 자신에게 발생한 비동기적 사건을 kernel에게 알리는 방법



System Call



System Call 추가하기 (실습)

- 목적
 - ✓ 기존 kernel에서 제공하지 않는 service를 user application에 제공
 - ➔ 새로운 system call을 만듦
- 단계
 - ✓ 커널 수정
 - System call 번호 할당
 - System call 호출 테이블 등록
 - System call 호출 함수 구현
 - Kernel 컴파일 및 target board 적재
 - ✓ User application 제작
 - 새로 구현한 system call을 사용하는 application 제작
 - Library 작성 (opt.)
 - Target board에 적재



E N D

