**운영체제 과제**

**유혁 교수님**

**2017320122 김정규**

setjmp.h는 "non-local jump"을 제공하기 위해 C 표준 라이브러리에 정의된 헤더로, 일반적인 서브루틴 호출과 리턴 시퀀스에서 분리된 제어 흐름이다. setjmp와 longjmp는 상호 보완적으로 작동하며 이 기능을 제공한다.

일반적으로 setjmp과 longjmp는 프로그램이나 스레드의 상태를 재설정하는 예외적인 메커니즘을 구현하는 데에 사용되는데, 여기에는 longjmp의 기능이 핵심적이다. 그와 달리 setjmp은 일반적으로 코루틴과 유사한 구문을 만드는 데에 사용된다.

**int setjmp(jmp\_buf env)**

로컬 jmp\_buf 버퍼를 설정하고 점프를 위해 초기화한다. 이 루틴은 프로그램의 호출 환경을 longjmp가 나중에 사용할 수 있도록 env 인수에 의해 지정된 환경 버퍼에 저장한다. 직접 호출에서 반환된 경우 setjmp는 0을 반환한다. 콜에서 longjmp로의 리턴인 경우 setjmp는 0이 아닌 값을 반환한다.

**void longjmp(jmp\_buf env, int value)**

동일한 프로그램 호출에서 setjmp 루틴을 호출하여 저장한 환경 버퍼인 env의 컨텍스트를 복원한다. 중첩된 신호 처리기에서 longjmp를 호출하는 것은 정의되지 않으며 값으로 지정한 값은 longjmp에서 setjmp로 전달된다. longjmp가 완료된 후, 프로그램 실행은 setjmp의 호출이 방금 돌아온 것처럼 계속된다. longjmp에 전달된 값이 0이면 setjmp는 1을 반환한 것처럼 동작하고, 그렇지 않으면 다른 지정된 값을 반환한 것으로 동작한다.

setjmp는 프로그램 실행 중 특정 시점에서의 환경(프로그램 상태)을 longjmp을 통해 어떤 이후 지점에서 사용할 수 있는 플랫폼별 데이터 구조(jmp\_buf)에 저장하여 setjmp에 의해 저장된 프로그램 상태를 jmp\_buf로 리턴한다. 이 과정은 setjmp가 환경을 살린 프로그램 실행 지점으로 돌아가는 'jump'라고 생각할 수 있다. 이렇게 컴퓨터에서 setjmp와 longjmp를 통해 예외적인 시그널을 송수신하는 signal handling이 일어나게 되는 것이다.