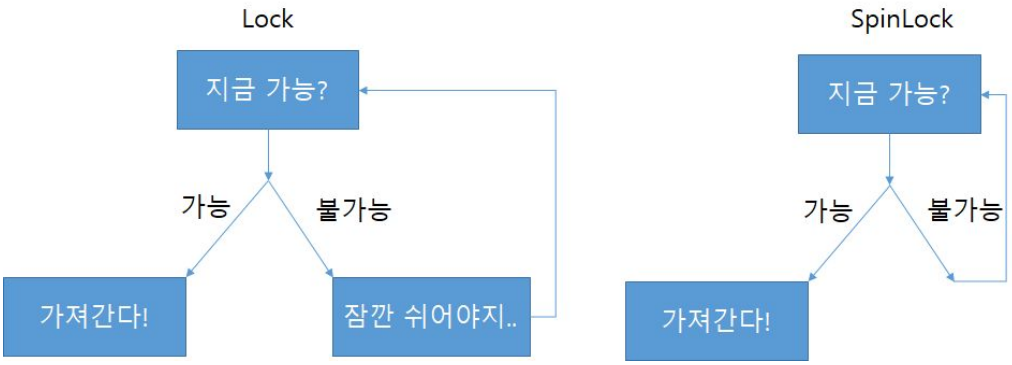
디지털컨버전스 기반 UXUI Front 전문 개발자 양성과정

강사 - Innova Lee(이상훈)
gcccompil3r@gmail.com
학생 - JungHyun LEE(이정현)
akdl911215@naver.com

스핀락(Spinlock)란 무엇인가?
스핀락은 Critical Section에 진입이 불가능할 때 진입이 가능할 때까지 루프를 돌면서 재시도 하는 방식으로 구현된 락을 가리킨다. 스핀락은 바쁜 대기의 한 종류이다.

스핀락은 운영체제의 스케줄링 지원을 받지 □□□기 때문에, 해당 스레드에 대한 Context Switch가 일어나지 않는다. 따라서 스핀락은 임계구역에 짧은 시간안에 진입할 수 있는 경우에 Context Switch을 제거할 수 있어서 효율적이다. 하지만 만약 스핀락이 오랜 시간을 소요한다면 다른 스레드를 실행하지 못하고 대기하게 되며, 이 경우 비효율적인 결과를 가져온다.



컨텍스트 스위칭(Context Switching)
CPU내에 존재하는 레지스터들은 현재 실행중인 프로세스 관련 데이터들로 채워집니다. 실행중인 프로세스가 변경이 되면, CPU내 레지스터들의 값이 변경되어야 하는데, 변경되기 전에 이전 프로세스가 지니고 있던 데이터들을 어딘가에 저장해 주어야 한다. 그리고 새로 실행되는 프로세스가 아니라면 이전에 실행될 때 레지스터들이 지니고 있던 데이터를 불러와서 이어서 실행해야 한다. 이 과정이 컨텍스트 스위칭이다. 실행되는 프로세스의 변경 과정에서 발생하는 컨텍스트 스위칭은 시스템에 많은 부담을 준다. 레지스터 개수가 많을수록, 프로세스별로 관리되어야할 데이터 종류가 많을수록 더 부담이 된다. 컨텍스트 스위칭에 소요되는 시간을 줄이려면 저장하고 복원하는 컨텍스트 정의 개수를 줄여주면 된다.

.

인터럽트란?
운영체제에서 컴퓨터에 예기치 않은 일이 발생하더라도 작동이 중단되지 않고 계속적으로 업무처리를 할 수 있도록 해주는 기능

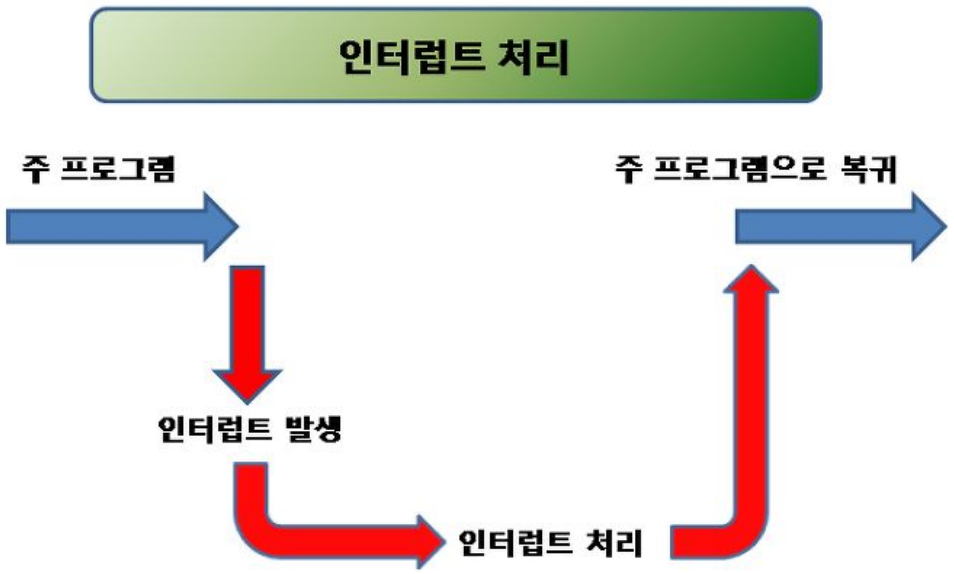
마이크로프로세서에서 인터럽트란 마이크로프로세서(CPU) 가 프로그램을 실행하고 있을 때, 입출력 하드웨어 등의 장치에 예외상황이 발생하여 처리가 필요한 경우 마이크로프로세서에게 알려 처리할 수 있도록 하는 것을 말합니다.

CPU가 프로그램을 실행하고 있을 때, 입출력 하드웨어 등의 장치나 예외상황이 발생하여 처리가 필요할 경우에 마이크로프로세서에게 알려 처리할 수 있도록 하는 것을 말한다.
인터럽트는 크게 하드웨어 인터럽트와 소프트웨어 인터럽트로 나뉜다.

-하드웨어 인터럽트
하드웨어가 발생시키는 인터럽트로, CPU가 아닌 다른 하드웨어 장치가 cpu에 어떤 사실을 알려주거나 cpu 서비스를 요청해야 할 경우 발생시킨다.

-소프트웨어 인터럽트
소프트웨어가 발생시키는 인터럽트입니다. 소프트웨어(사용자 프로그램)가 스스로 인터럽트 라인을 세팅한다.
종류 : 예외 상황, system call

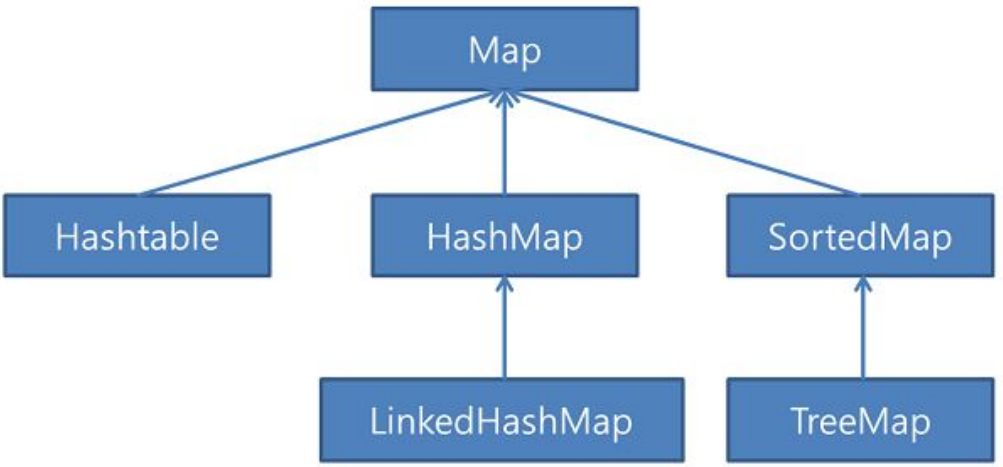
인터럽트를 발생시키기 위해 하드웨어/소프트웨어는 cpu내에 있는 인터럽트 라인을 세팅하여 인터럽트를 발생시킨다.cpu는 매번 명령을 수행하기 전에 인터럽트라인이 세팅되어있는지를 검사한다



해쉬맵(HashMap)
해쉬맵을 사용하면 데이터 저장은 느리지만 많은 양의 데이터를 검색하는데 있어서 뛰어납니다. HashMap은 Map을 구현했으므로 키(Key)와 값(Value)을 묶어서 하나의 데이터(entry)로 저장합니다.

키는 중복 허용이 되지 않고, 값은 중복이 허용됩니다. 예를 들어서 ID는 중복이 안되지만, 여러명의 사람이 비밀번호를 1234라고 저장하는 경우가 있다고 생각하면 좋습니다.

해쉬맵이란 Map인터페이스 중 하나로써, 키와 밸류값으로 묶어 데이터를 저장하는자료구조입니다. Hashing을 사용하므로써 많은양의 데이터를 검색하는데 뛰어난 성능을 가지고 있습니다.



Process Control Block (PCB)

프로세스 제어 블록은 특정한 프로세를 관리할 필요가 있는 정보를 포함하는, 운영체제 커널의 자료구조입니다. PCB는 운영 체제가 프로세스를 표현한 것이라 할 수 있습니다.

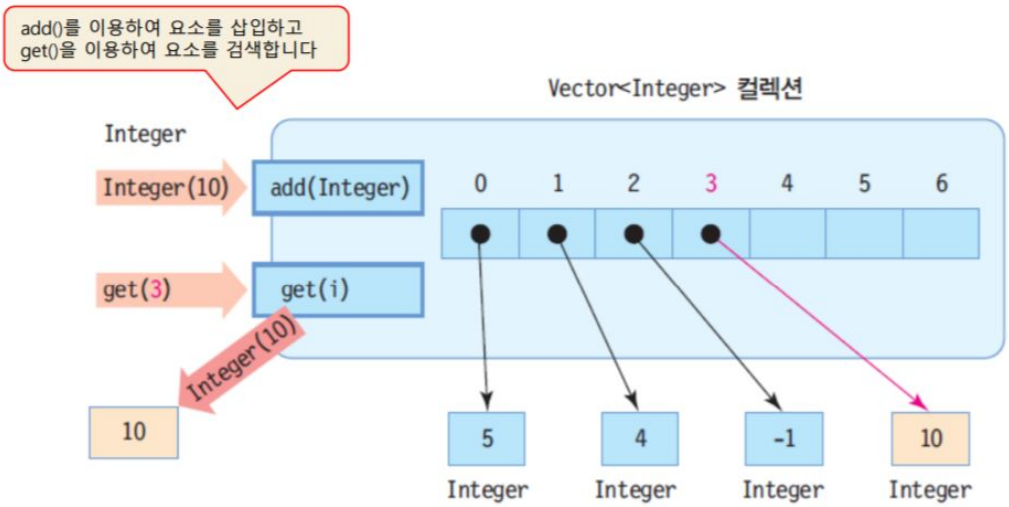
운영체제가 프로세스 스케줄링을 위해 프로세스에 관한 모든 정보를 가지고 있는 데이터베이스를 PCB라고 합니다.

운영체제에서 프로세스는 PCB로 나타내어지며, PCB는 프로세스에 대한 중요한 정보를 가지고 있는 자료입니다. 각 프로세스가 생성될 때마다 고유의 PCB가 생성되고, 프로세스가 완료되면 PCB는 제거됩니다.

프로세스는 CPU를 점유하여 작업을 처리하다가도 상태가 전이되면, 진행하던 작업 내용들을 모두 정리하고 CPU를 반환해야 하는데, 이때 진행하던 작업들을 모두 저장하지 않으면 다음에 자신의 순서가 왔을 때 어떠한 작업을 해야하는지 알 수 없는 사태가 발생합니다.

따라서 프로세스는 CPU가 처리하던 작업의 내용들을 자신의 PCB로부터 해당 정보들을 CPU에 넘겨와서 계속해서 하더너 작업을 진행할 수 있게 됩니다.

Vector란?
Vector는 ArrayList와 동일한 내부구조를 가지고 있습니다.
ArrayList와 마찬가지로 Vector내부에 값이 추가되면 자동으로 크기가 조절되며 그다음 객체들은 한 자리씩 뒤로 이동됩니다.
하지만 모든 것이 다 똑같은 것은 아니고 Vector와 Arraylist의 한가지 다른 점이 있는데 Vector는 동기화된 메소드로 구성되어 있기 때문에 멀티 스레드가 동시에 이 메소드를 실행할 수 없고, 하나의 스레드가 실행을 완료해야만 다른 스레들이 실행할 수 있습니다. 그래서 멀티 스레드 환경에서 안전하게 객체를 추가하고 삭제할 수 있습니다.



Vector의 단점 (ArrayList와의 비교)
벡터는 항상 동기화되는 장점이자 단점을 가지고 있습니다.
스레드가 1개일때도 동기화를 하기 때문에 ArrayLsit보다 성능이 떨어집니다. Arraylist는 기본적인 기능은 벡터와 동일하나 자동 동기화 기능이 빠져있고, 동기화 옵션이 존재합니다. 그리고 벡터에 비해 속도가 더 빠르기 때문에 벡터에 비해 많이 쓰이고 있습니다.