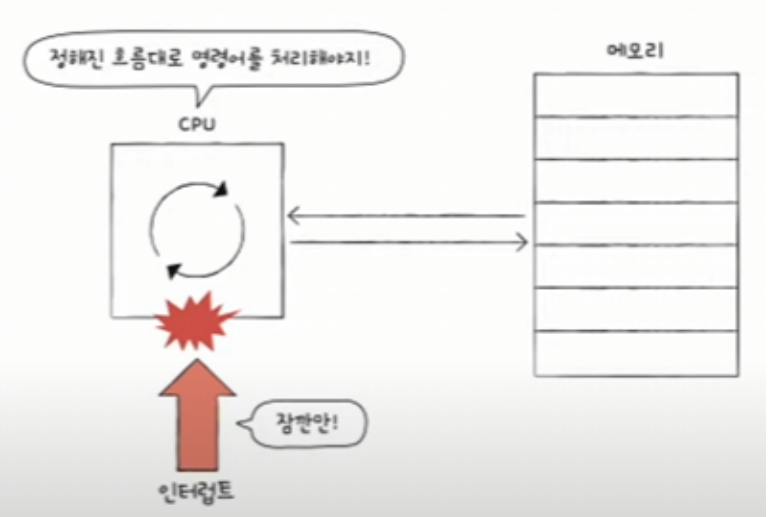
11강. 명령어 사이클과 인터럽트

명령어 사이클

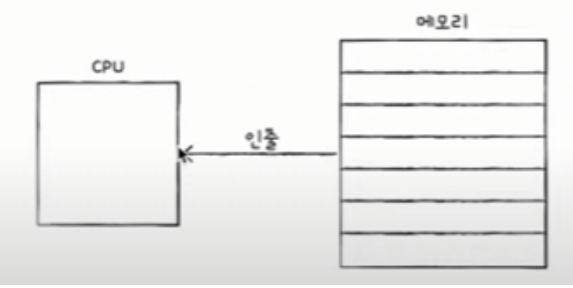


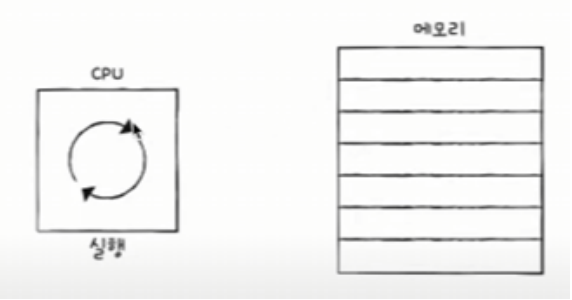
프로그램 속 명령어들은 일정한 주기가 반복되며 실행합니다. 더 자세히 살펴보면, 프로그램 속 명령어는 “가져오기 → 실행하기” 의 흐름으로 반복됩니다.

이 한 단위를 **명령어 사이클**이라고 합니다. 즉, CPU는 항상

1. 메모리에서 명령어를 가져오고 (Fetch) → 인출
2. 그 명령어를 해석하고 실행(Execute)

하는 주기를 반복합니다.





인출 사이클 (Fetch Cycle)

메모리에 저장된 명령어를 CPU로 가져오는 목적을 지닙니다.

인출 사이클의 과정은 다음과 같습니다.

1. PC(프로그램 카운터)가 실행할 명령어의 주소를 가르킵니다.
2. 제어장치가 메모리에 읽기(Read) 신호를 보냅니다.
3. 해당 주소의 명령어가 명령어 레지스터(IR)로 전달됩니다.
4. PC(프로그램 카운터)는 자동으로 다음 명령어를 가르키도록 증가됩니다.

실행 사이클 (Execute Cycle)

가져온 명령어를 실제로 실행하는 목적을 지닙니다.

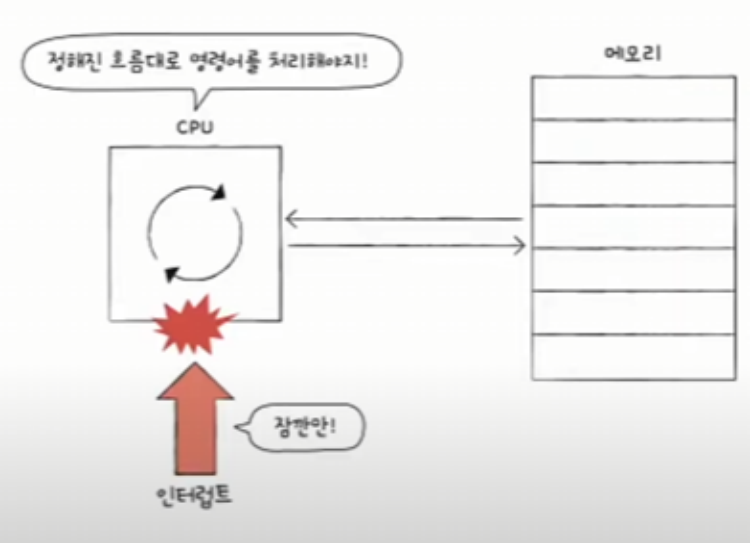
실행 사이클의 과정은 다음과 같습니다.

1. 제어장치가 IR의 명령어를 해석합니다.
2. ALU, 레지스터, 메모리, 입출력 장치에 적절한 제어 신호를 보냅니다.
3. 연산이 수행되고 결과가 레지스터나 메모리에 저장됩니다.
4. 결과에 따라 플래그가 갱신될 수도 있습니다.

인터럽트

인터럽트는 CPU가 명령어를 차례대로 실행하던 중, 예상치 못한 상황이나 특별한 요청이 발생하면 신호를 보냅니다. 이 신호를 인터럽트라고 부릅니다.

인터럽트가 발생하면 CPU는 현재 작업을 중단하고, 해당 요청을 처리한 뒤 다시 원래 작업으로 돌아갑니다.



인터럽트의 종류

동기 인터럽트 (Synchronous Interrupt)

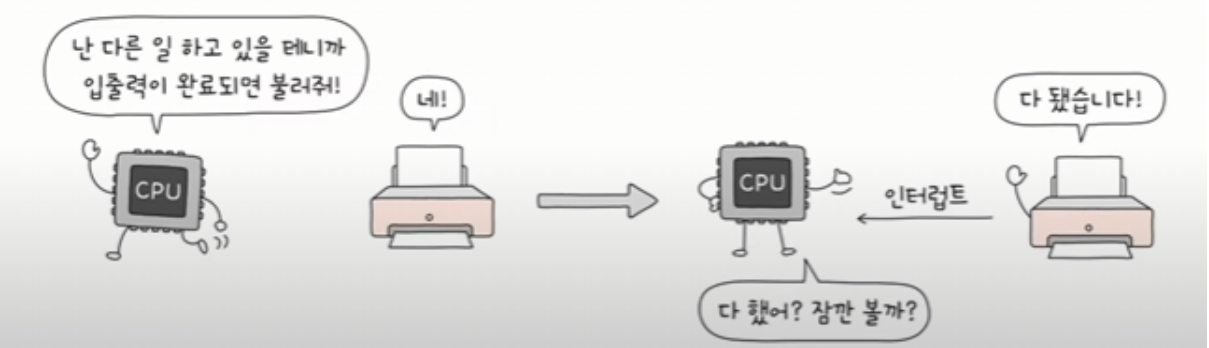
명령어 실행 결과에 의해 발생하는 인터럽트입니다. CPU가 특정 명령어를 실행하다가, 그 명령어 때문에 문제가 생길 때 바로 발생합니다.

예시

- 0으로 나누기
- 잘못된 명령어 실행
- 페이지 폴트
- 산술 오버플로우

비동기 인터럽트 (Asynchronous Interrupt)

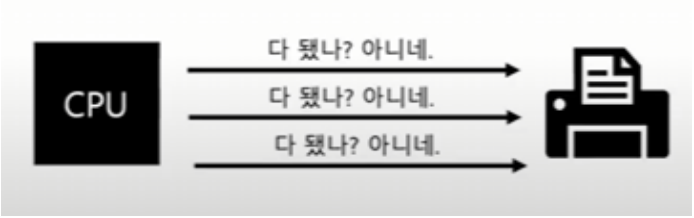
CPU가 실행하는 명령어와 무관하게, 외부 요인 때문에 발생하는 인터럽트입니다. 보통 언제 발생할지 예측 불가능 하고, 주로 하드웨어 장치에서 발생하는 경우가 많습니다.



비동기 인터럽트(하드웨어 인터럽트)

입출력 작업 도중에도 효율적으로 명령어를 처리하기 위해 하드웨어 인터럽트를 사용합니다.

입출력 장치는 CPU보다 느리기 때문에, 만약 인터럽트가 없다면 CPU는 프린트 완료 여부를 수시로 확인할 것입니다.

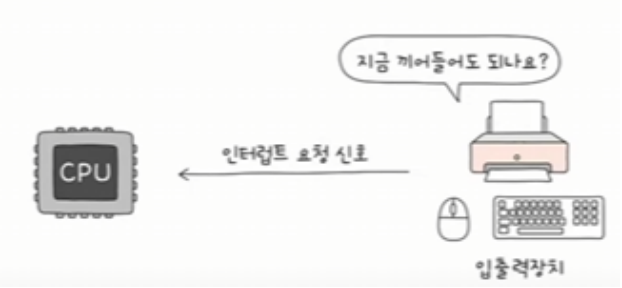


하드웨어 인터럽트의 처리 순서

인터럽트의 종류와 관계없이 인터럽트 처리 순서는 비슷합니다.

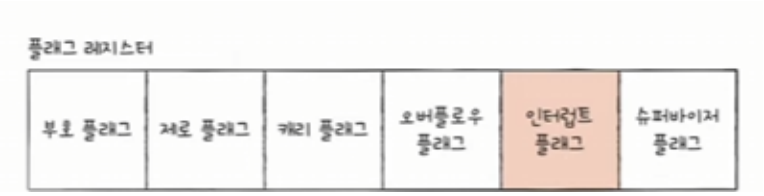
1. 입출력 장치는 CPU에 인터럽트 요청 신호를 보냅니다.

- 인터럽트 요청 신호



2. CPU는 실행 사이클이 끝나고 명령어를 인출하기 전 항상 인터럽트 여부를 확인합니다.

3. CPU는 인터럽트 요청을 확인하고 인터럽트 플래그를 통해 현재 인터럽트를 받아들일 수 있는 지 여부를 확인합니다.



- 모든 인터럽트를 인터럽트 플래그로 막을 수 있는 것은 아닙니다.

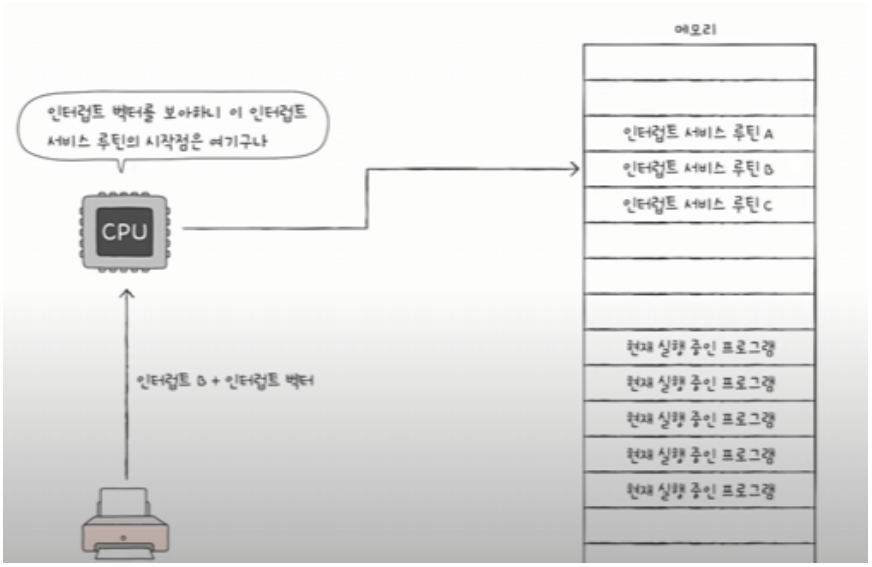
  하드웨어 인터럽트에서는 플래그로 막을 수 있는, 막을 수 없는 인터럽트가 있습니다.

4. 인터럽트를 받아들일 수 있다면 CPU는 지금까지의 작업을 백업합니다.

5. CPU는 인터럽트 벡터를 참조하여 인터럽트 서비스 루틴을 실행합니다.

- 인터럽트 벡터

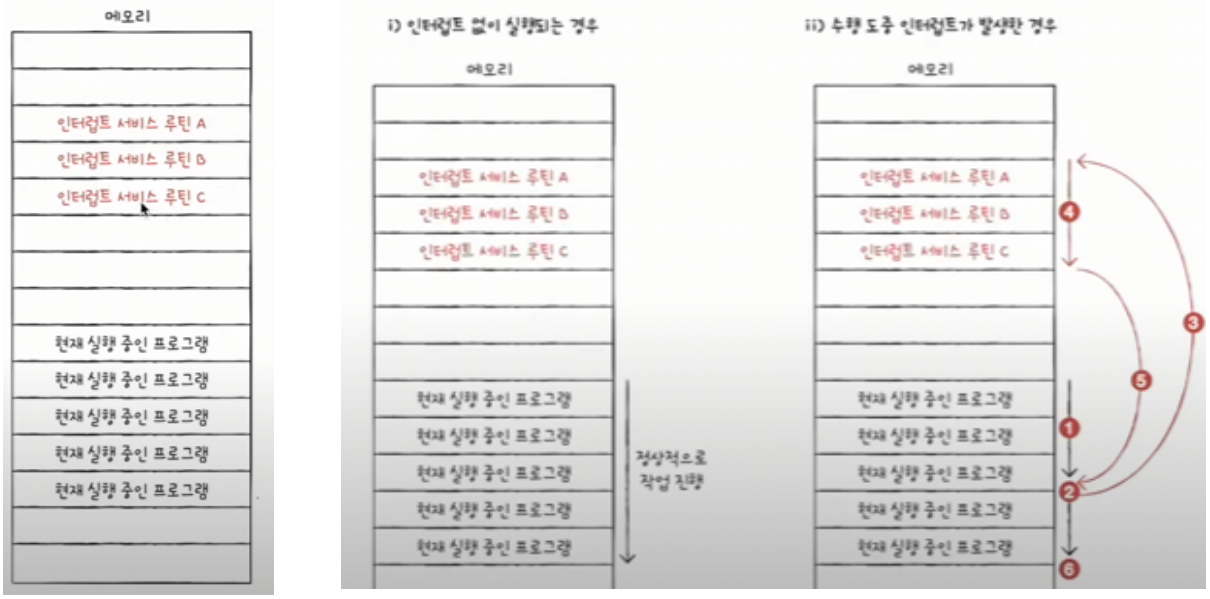
  각각의 인터럽트를 구분하기 위한 정보



- 인터럽트 서비스 루틴(Interrupt Service Routine, ISR)

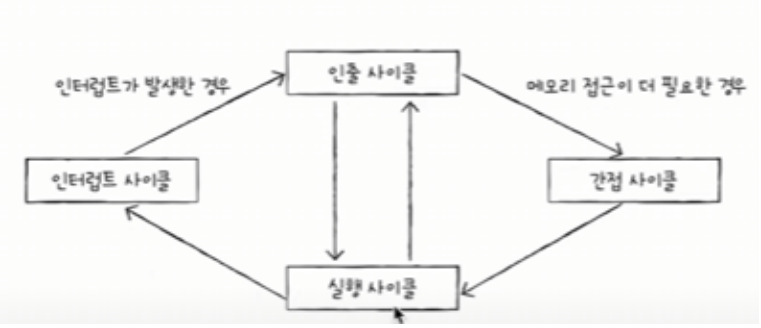
  인터럽트가 발생했을 때, 그 상황을 처리하기 위해 실행되는 작은 프로그램입니다.

  보통 운영체제(OS) 내부에 미리 준비되어 있으며, 인터럽트 종류 별로 각각의 루틴이 존재합니다.



6. 인터럽트 서비스 루틴 실행이 끝나면 4번에서 백업해 둔 작업을 복구하여 실행을 재개합니다.

이를 통해 전체 명령어 사이클를 그래프로 표현하면 다음과 같습니다.



- 간접 사이클(Indirect Cycle)

  어떤 명령어는 피연산자(Operand)가 바로 주어지지 않고, 피연산자의 주소가 저장된 위치를 먼저 알려줍니다.

  즉, 한 번 더 메모리에 접근해서 실제 피연산자를 가져와야 합니다. 추가적인 메모리 접근 단계를 간접 사이클이라고 합니다.