12주 2강

인터럽트 처리

인터럽트 처리

- 인터럽트(interrupt)의 어원상 의미 : 끼어들기
  - 시스템 내의 한 장치가 인터럽트를 발생하면 CPU는 다른 장치의 일을 잠시 중단하고 작업 상태를 기억장치에 저장.
  - 인터럽트를 발생시킨 장치의 작업을 우선적으로 진행.
- 인터럽트의 개념
  - CPU가 현재 실행 중인 프로그램을 강제적으로 중단하고, 특정 주소에 위치한 새로운 프로그램을 수행하게 하는 것
  - 현재 실행 중인 프로그램의 중요 데이터는 주기억장치에 저장,
  - 새 프로그램이 종료되면 다시 프로그램을 진행한다.
- 인터럽트 서비스 루틴(ISR, Interrupt Service Routine)
  - 인터럽트를 처리하기 위해 실행되는 새로운 프로그램 루틴

인터럽트에 의한 제어이동

① 인터럽트가 발생하면 CPU는 어떤 장치가 요구했는지 확인

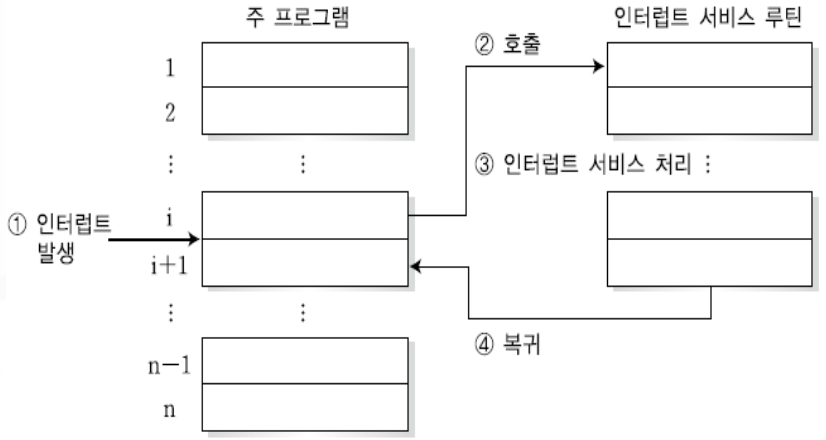
- PC에 저장된 현재 프로그램의 다음 실행 명령어의 주소를 스택(stack)에 저장한다.

② 인터럽트 서비스 루틴을 호출하기 위하여 그 루틴의 시작 주소를 PC에 적재

- 시작 주소는 인터럽트를 요구한 장치에서 전송되거나 미리 정해진 값으로 결정된다.

③ 인터럽트 서비스를 마지막까지 순차적으로 실행한다.

④ 중단되었던 기존 프로그램의 실행을 계속하도록 주 프로그램으로 복귀한다



인터럽트의 발생 원인과 종류

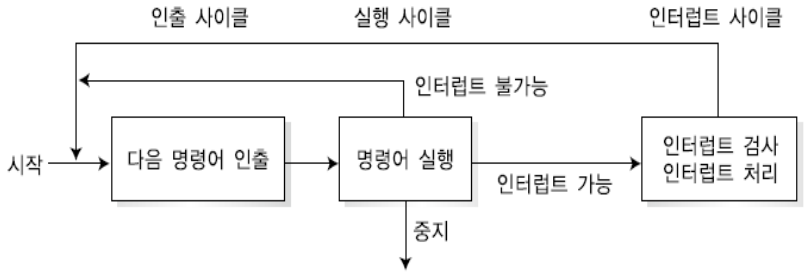
- 기계 착오 인터럽트
  - 프로그램 실행 중 정전이나 컴퓨터 내에서 기계적인 문제가 발생한 경우 발생
- 슈퍼바이저 호출 인터럽트(Superviser Call Interrupt)
  - 사용자가 입출력과 같은 서비스를 받기 위해 슈퍼바이저 호출(SVC) 명령어를 사용하여 운영체제에 서비스를 요청할 때 발생
- 외부 인터럽트(External Interrupt)
  - 오퍼레이터나 타이머에 의해 의도적으로 프로그램이 중단된 경우 발생
- 입출력 인터럽트(I/O Interrupt)
  - 입출력의 종료나 입출력의 오류에 의해 CPU의 기능이 요청되는 경우 발생
- 프로그램 검사 인터럽트(Program Check Interrupt)
  - 프로그램 실행도중 불법적 명령 수행 같은 프로그램의 문제가 발생한 경우 발생
- 재시작 인터럽트(Restart Interrupt)
  - 오퍼레이터 및 다른 프로세서에 의해서 재시작 명령이 도착하였을 때 발생

컴퓨터에서 인터럽트의 발생과 처리 예

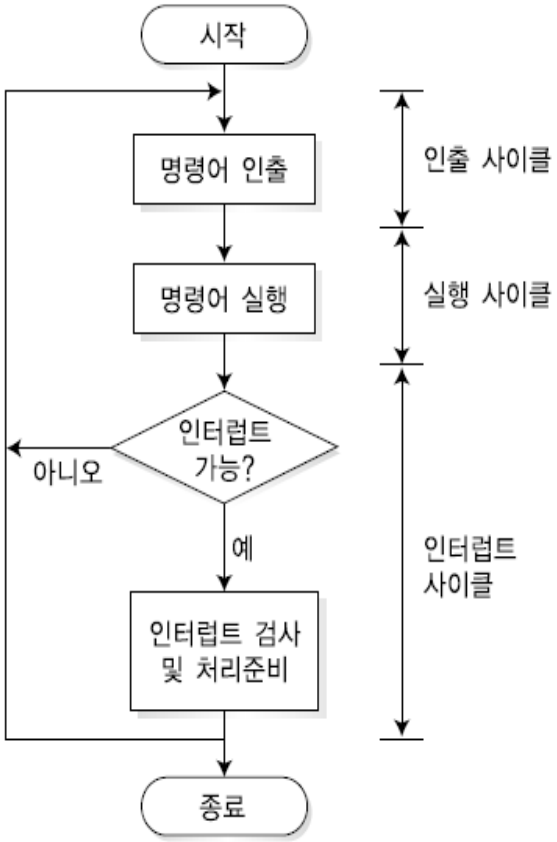
- 프로그램에 의한 발생
  - 프로그램 잘못 실행으로 오버플로우, 0으로 나누기 등이 발생하면 프로그램이 종료되는 인터럽트 발생
- 컴퓨터 내부 하드웨어에 의한 발생
  - 하드웨어 결함의 예로는 기억장치 패리티 오류 발생이 있다.
  - 기억장치에 저장된 데이터에 오류가 발생한 것으로 CPU는 해당 데이터의 읽기 동작을 중지하는 인터럽트가 발생
- I/O장치에 의한 발생
  - I/O 제어기에 의해 프린터가 출력되거나, 키보드의 입력으로 중앙처리장치의 정상처리 과정이 인터럽트되는 경우
- 인터럽트 처리 예
  - 오버플로우에 의한 인터럽트: 오버플로우 발생 경고 메시지와 주 프로그램을 종료하는 인터럽트 서비스 루틴이 동작
  - 프린터 제어기에 의한 인터럽트: 프린트의 출력을 동작하는 인터럽트 서비스 루틴이 수행되고 동작 종료 후, 주 프로그램으로 되돌아 간다

인터럽트 사이클

- 인터럽트 사이클(Interrupt Cycle)
  - 중앙처리장치가 인터럽트 요구가 있는지의 검사를 수행하는 과정
  - 인터럽트 발생이 없다면, 다음 명령어를 인출하는 인출 부 사이클이 수행
  - 인터럽트 요구가 대기 중이라면 인터럽트 사이클에 의해서, 현재 프로그램의 실행을 중단하고 프로그램 상태(program state)를 저장
  - PC를 인터럽트 처리 루틴의 시작 주소로 세트하고 인터럽트를 처리
- 명령어 사이클
  - 인출 사이클, 실행 사이클, 인터럽트 사이클 세 개의 부 사이클로 구성
- 인터럽트 사이클을 포함한 명령어 사이클



인터럽트 부 사이클이 포함된 명령어 사이클의 순서도



인터럽트 사이클의 마이크로 연산

인터럽트의 동작

- 인터럽트 발생이 확인되면, 현재 진행 중인 주 프로그램의 복귀 정보를 주기억장치의 일부에 저장하고 인터럽트 서비스 루틴을 수행하는 것이다.

인터럽트의 마이크로 연산

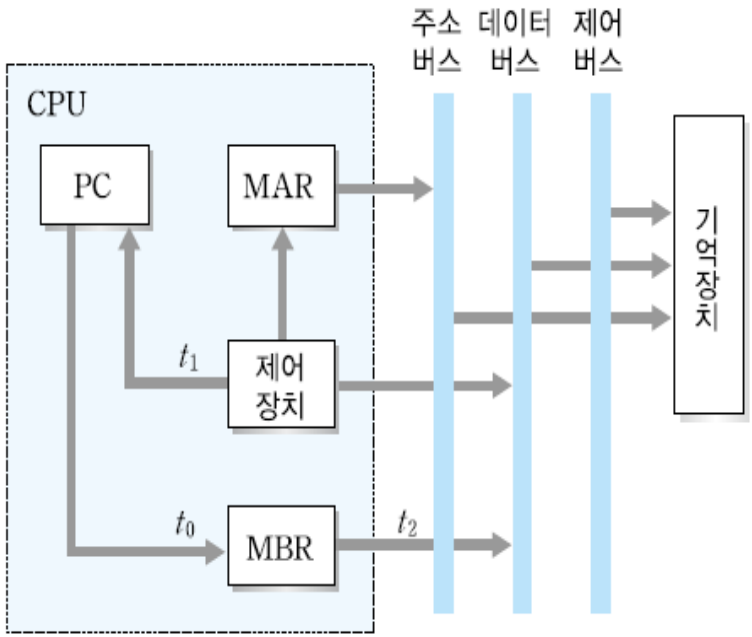
t_0 : MBR ← PC
t_1 : MAR ← SP, PC ← ISR 시작주소
t_2 : M[MAR] ← MBR

① 클럭 t_0에서는 PC의 내용이 MBR로 전송된다.
- PC는 다음 명령어 주소로 주 프로그램의 복귀 정보를 저장하는 과정이다.

② 클럭 t_1에서는 SP의 내용이 MAR로 전송되고, PC의 내용은 인터럽트 서비스 루틴(ISR)의 시작 주소로 변경된다.
- SP는 MBR에 저장되어 있는 내용을 스택에 저장하기 위해서 사용

③ 클럭 t_2에서는 MBR에 저장되어 있던 원래 PC의 내용이 스택에 저장된다.
- 주 프로그램의 복귀 주소를 주기억장치의 스택에 저장하는 과정이다.

인터럽트 부 사이클에서 데이터 흐름

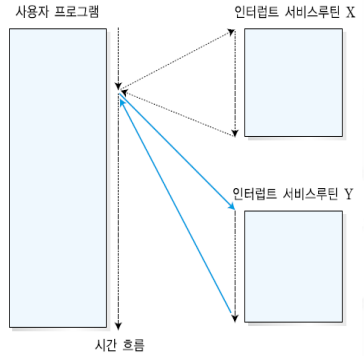


다중 인터럽트

- 인터럽트 서비스 루틴(ISR)을 수행하는 동안 또 다른 인터럽트가 발생하는 것.
  - 다중 인터럽트 처리방법은 인터럽트 불가능과 우선순위 인터럽트가 있다.

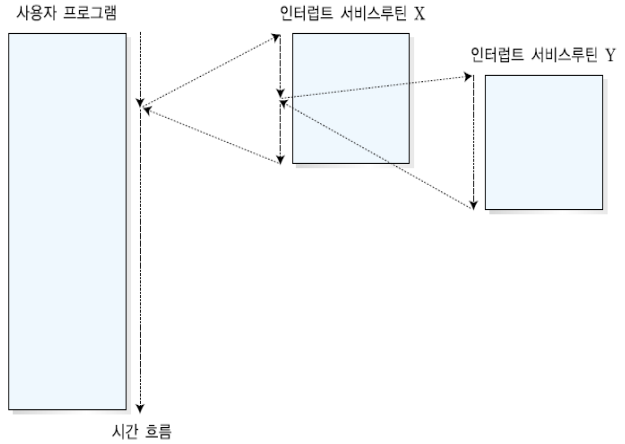
- 인터럽트 불가능(Interrupt Disabled)
  - ISR을 처리하고 있는 도중에는 새로운 인터럽트 처리요구가 들어오더라도 CPU가 인터럽트 사이클을 수행하지 않도록 방지하는 기능
  - 나중에 발생한 인터럽트를 무시하는 것이다. 그렇지만 대기하고 있다가 현재의 인터럽트에 대한 처리가 종료된 후에 발생한 순서대로 처리된다.

- 순차적인 다중 인터럽트 처리



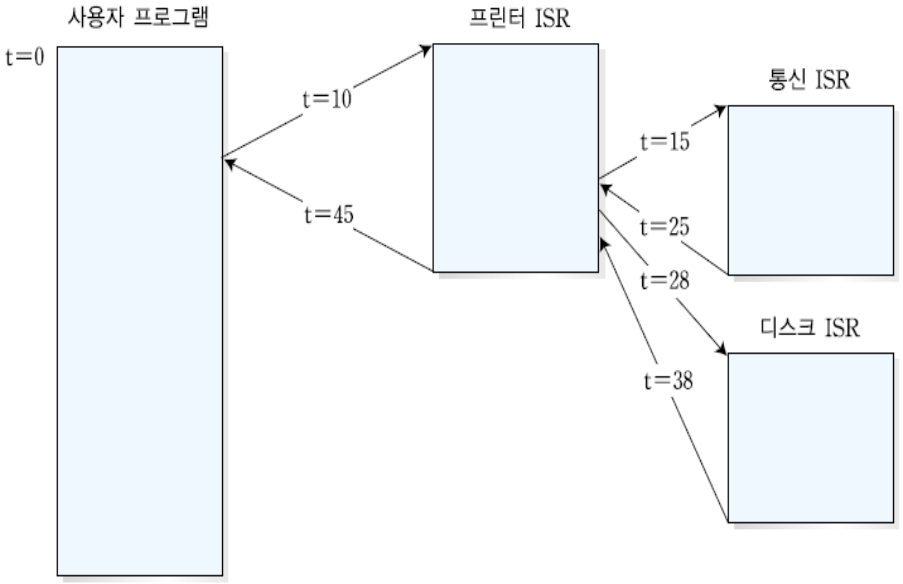
우선순위 인터럽트(Priority Interrupt)

- 인터럽트의 우선순위를 정하고, 우선순위가 낮은 인터럽트가 처리되고 있는 동안에 우선순위가 더 높은 인터럽트가 들어오면, 현재의 인터럽트 서비스 루틴의 수행을 중단하고 새로운 인터럽트를 처리하는 방법



- 우선순위 인터럽트 처리 과정
  - 인터럽트에 의해서 인터럽트 서비스 루틴 X가 실행된다.
  - 우선순위가 더 높은 인터럽트 요구가 들어오면, 서비스 루틴 X의 실행은 중지되고 인터럽트 서비스 루틴 Y가 수행된다.
  - 서비스 루틴 Y의 실행이 끝나면, 다시 서비스 루틴 X로 복귀한다.
  - 서비스 루틴 X가 종료되면 사용자 프로그램으로 복귀한다.

다중 인터럽트 처리에서 시간 흐름의 예



다음 시간

12주 3강. 명령어 파이프라이닝