



11강. 장치 관리

방송대 컴퓨터과학과
김진욱 교수



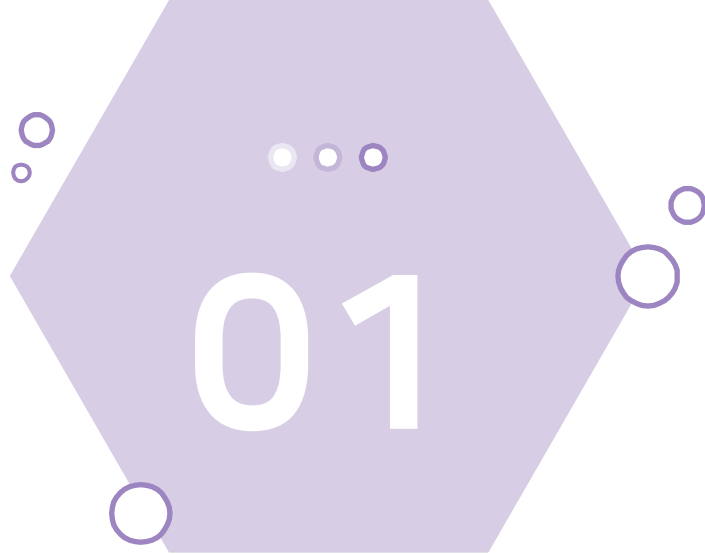
목차

01 장치의 개념

02 장치의 구성

03 입출력 처리 유형

04 입출력 관리



장치의 개념

컴퓨터 시스템의 구성

■ 다양한 장치들

- CPU, 메모리: 프로세스 실행에 필수
- 나머지 장치들: 프로세스 실행 시 데이터 입력이나 출력에 사용하는 입출력장치



장치의 구분

입출력장치의 구분

- 장치의 기능적 특징과 장치 관리자의 관리 방법에 따라 구분
- 세 가지 범주: 전용장치, 공용장치, 가상장치



입출력장치의 구분

■ 전용장치

- 한 번에 단지 하나의 프로세스에만 할당
- 예: 테이프 드라이브, 프린터, 플로터 등
- 단점: 대기시간이 길어질 수 있음

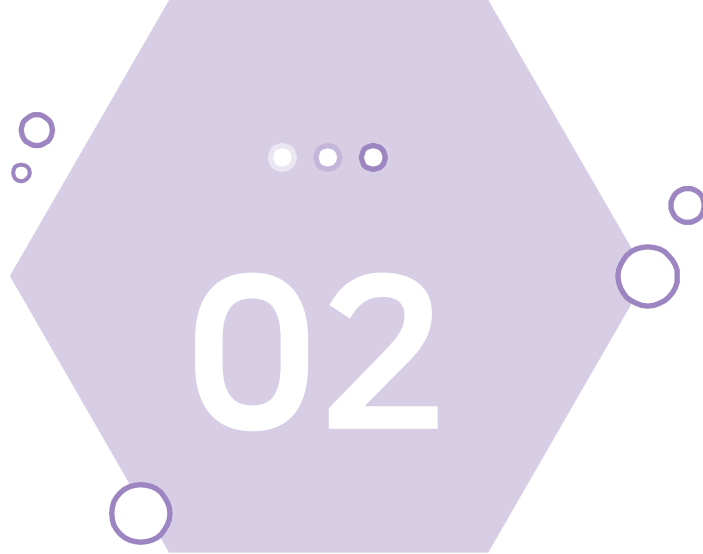
■ 공용장치

- 여러 프로세스에 동시에 할당
- 예: 디스크 같은 직접접근 저장장치
- 스케줄링 기법 필요

입출력장치의 구분

■ 가상장치

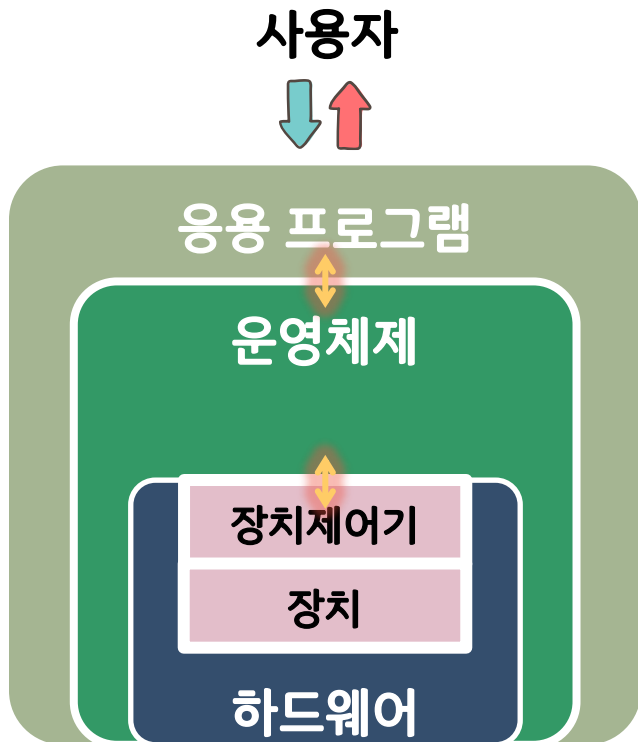
- 전용장치를 가상의 공용장치처럼 보이게 함
- 디스크 같은 공용장치를 이용
- 예: 스펠링을 적용한 플로터 등



장치의 구성

○ 장치의 구성

■ 논리적 구성



• 장치제어기

- » 장치를 직접적으로 다루는 전자장치
- » 장치에서 발생하는 각종 데이터를 전자적인 신호로 변환하여 운영체제로 보냄
- » 운영체제가 요청하는 명령을 받아 장치를 구동
- » 운영체제가 보내는 출력을 장치에 맞게 변환

○ 장치의 구성

■ 논리적 구성



• 장치 드라이버

- » 응용 프로그램의 입출력 요청을 해당 장치에 맞도록 변환
- » 장치의 종류나 제조사에 따라 장치제어기가 이해하는 명령이나 명령의 종류가 다를 수 있기 때문
- » 보통 장치 제조사에서 해당 장치의 드라이버도 같이 제공

○ 장치의 구성

■ 물리적 구성

• CPU의 장치 사용법

- » 장치제어기 내의 레지스터를 활용
- » 장치의 상태 확인(읽기), 장치에 명령(쓰기)
- » 통신: 입출력 명령, 메모리 사상 입출력



○ 장치의 구성

■ 물리적 구성

• CPU의 장치 사용법

- » 장치제어기 내의 레지스터를 활용
- » 장치의 상태 확인(읽기), 장치에 명령(쓰기)
- » 통신: 입출력 명령, 메모리 사상 입출력



○ 장치의 구성

■ 물리적 구성

• CPU의 장치 사용법

- » 장치제어기 내의 레지스터를 활용
- » 장치의 상태 확인(읽기), 장치에 명령(쓰기)
- » 통신: 입출력 명령, 메모리 사상 입출력

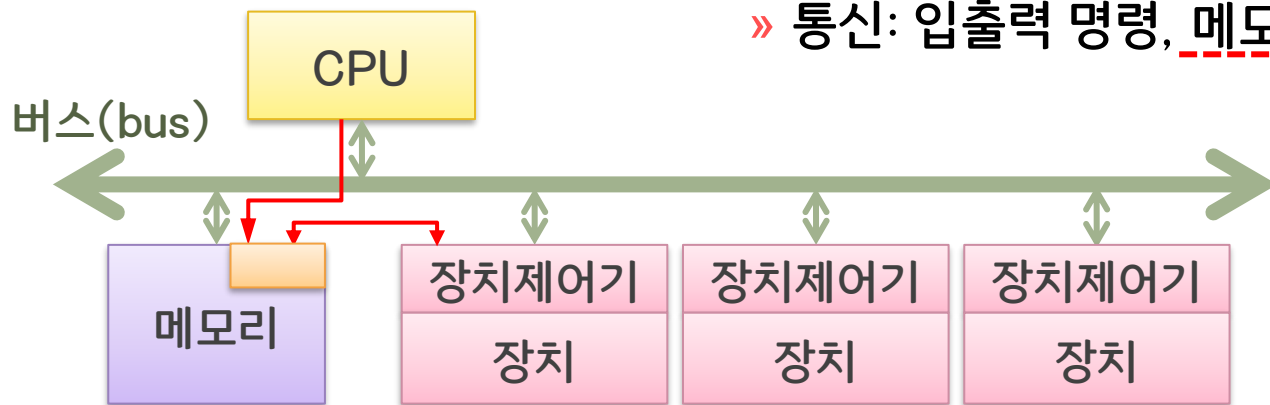


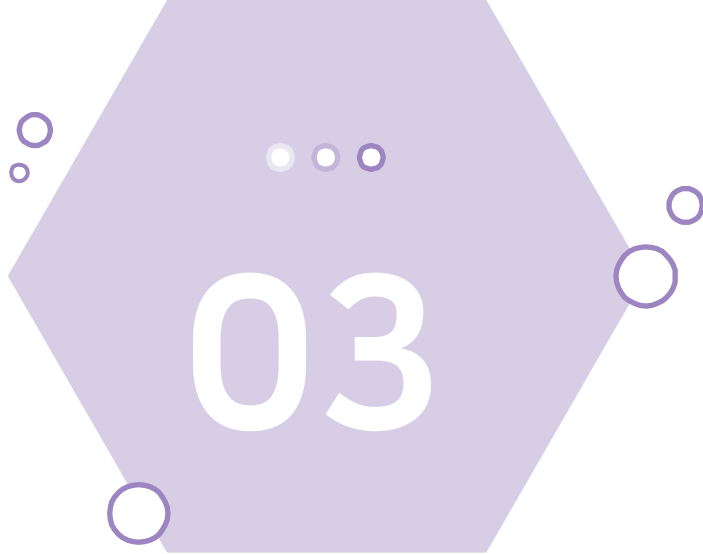
○ 장치의 구성

■ 물리적 구성

• CPU의 장치 사용법

- » 장치제어기 내의 레지스터를 활용
- » 장치의 상태 확인(읽기), 장치에 명령(쓰기)
- » 통신: 입출력 명령, 메모리 사상 입출력





입출력 처리 유형

입출력 처리 유형

■ 입출력 처리

- 프로세스가 진행하며 입출력이 발생하는 경우
- 세 가지 유형: 프로그램 방법, 인터럽트 방법, DMA 방법

입출력 처리 유형

■ 프로그램 방법

- CPU만을 이용하여 입출력을 처리
- 폴링 이용
- 예:

```
repeat
```

```
while State(프린터) != 가용 do skip;
```

```
한 줄 인쇄;
```

```
until 10줄 인쇄 완료;
```

- 단점: 비효율적 - CPU의 낭비가 심함

★ 폴링(polling)

CPU가 입출력장치의 상태를 지속적으로 확인하며 원하는 상태가 될 때까지 기다리는 것

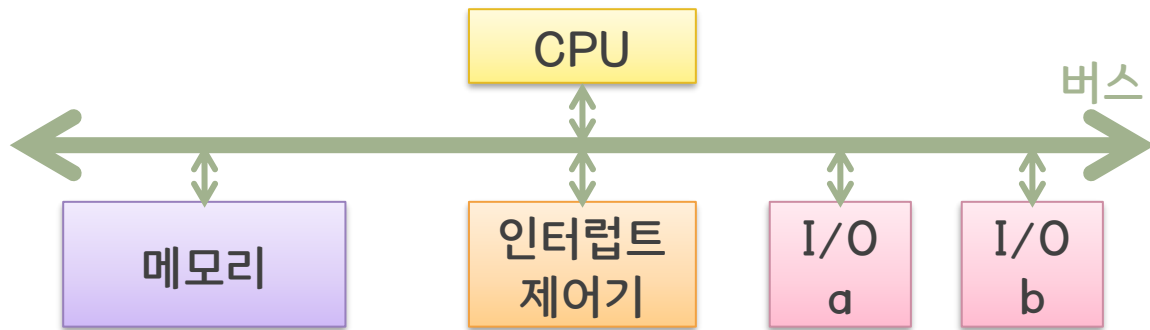
입출력 처리 유형

■ 인터럽트 방법

- 인터럽트를 이용하여 입출력을 처리
- 프로세스를 대기상태로 보내고 인터럽트가 발생할 때까지 CPU는 다른 작업을 하면 됨

★ 인터럽트(interrupt)

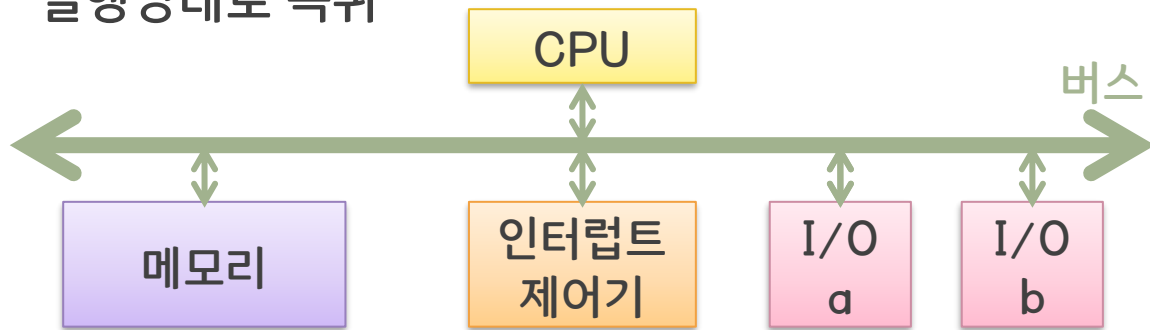
어떤 장치가 다른 장치의 작업을 잠시 중단시키고 자신의 상태를 알리는 기능



입출력 처리 유형

■ 인터럽트 방법 – 처리과정

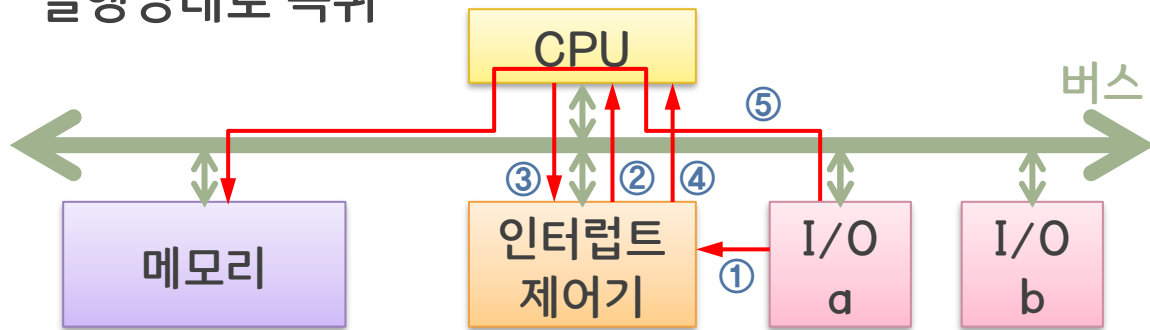
- ① 입출력장치가 가용한 상태가 되었다고 인터럽트 제어기에 신호를 보냄
- ② 인터럽트 제어기는 CPU에 인터럽트 신호를 보냄
- ③ CPU는 현재 실행 중이던 명령만 마친 후 즉시 인터럽트에 응답
- ④ 인터럽트 제어기는 이벤트 대상에 대한 정보를 CPU에게 보냄
- ⑤ CPU는 현재 상태를 보관하고 필요한 입출력 처리를 한 후 원래 프로세스 실행상태로 복귀



입출력 처리 유형

■ 인터럽트 방법 – 처리과정

- ① 입출력장치가 가용한 상태가 되었다고 인터럽트 제어기에 신호를 보냄
- ② 인터럽트 제어기는 CPU에 인터럽트 신호를 보냄
- ③ CPU는 현재 실행 중이던 명령만 마친 후 즉시 인터럽트에 응답
- ④ 인터럽트 제어기는 이벤트 대상에 대한 정보를 CPU에게 보냄
- ⑤ CPU는 현재 상태를 보관하고 필요한 입출력 처리를 한 후 원래 프로세스 실행상태로 복귀



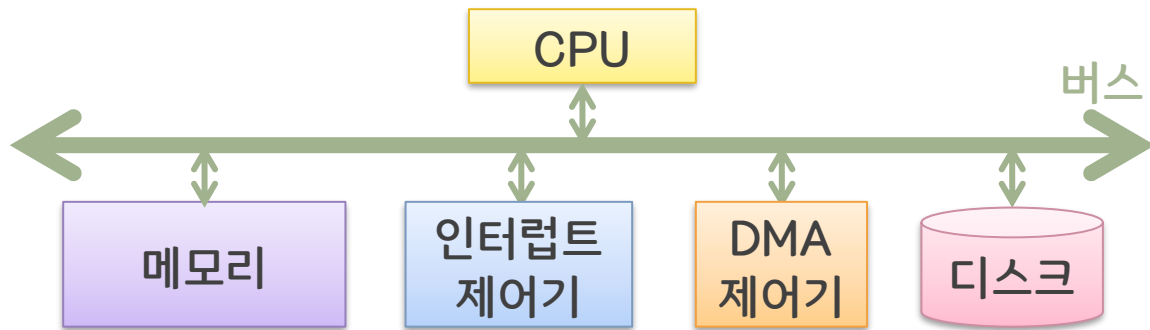
입출력 처리 유형

■ DMA 방법

- DMA 제어를 이용하여 입출력을 처리
- 프로세스를 대기상태로 보내고
인터럽트가 발생할 때까지 CPU는 다른 작업을 하면 됨
- 한 번에 입출력 양이 많은 경우에도 한 번의 인터럽트만 발생
(CPU의 효율 증대)

★ DMA(Direct Memory Access)

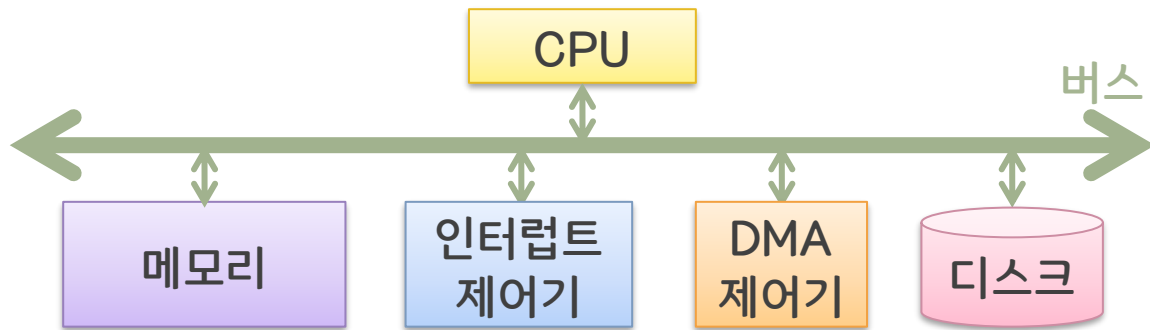
CPU를 통하지 않고 직접 메모리에 접근



입출력 처리 유형

■ DMA 방법 – 처리과정

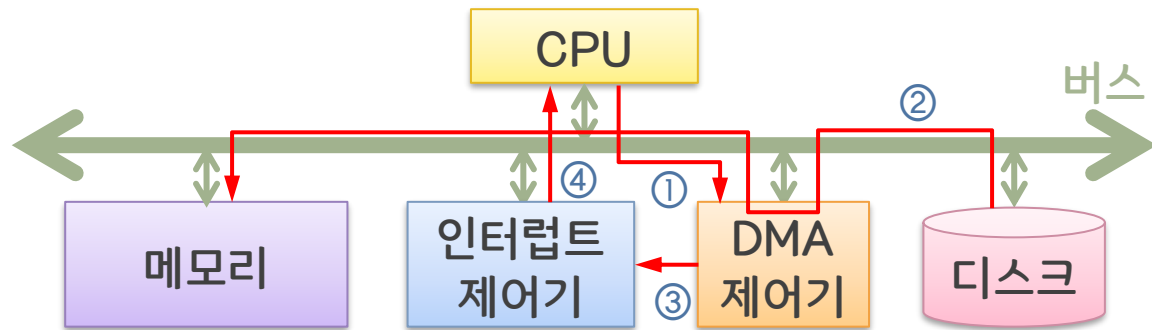
- ① CPU는 입출력에 필요한 정보를 DMA 제어기에 넘김
- ② [반복]DMA 제어기는 소스에서 목적지로 데이터를 보내도록 장치제어기에 요청
- ③ 원하는 양의 입출력이 끝나면 DMA 제어기는 인터럽트 제어기에 신호를 보내어
- ④ 인터럽트를 발생시켜 CPU에게 입출력 작업이 모두 끝났음을 알림



입출력 처리 유형

■ DMA 방법 – 처리과정

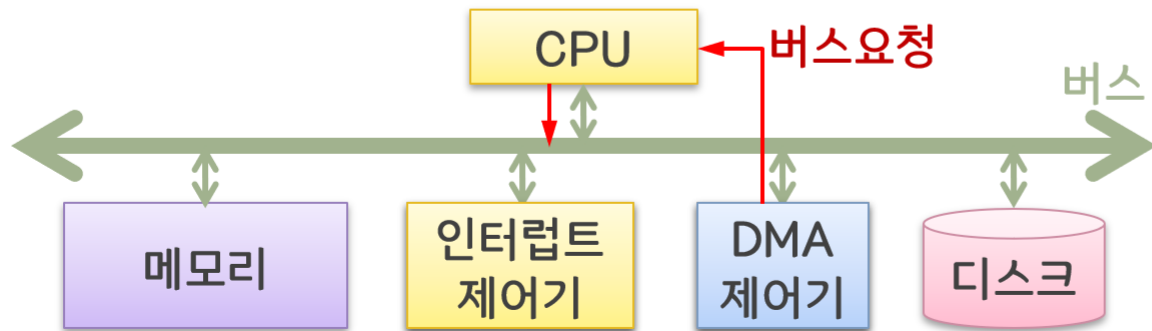
- ① CPU는 입출력에 필요한 정보를 DMA 제어기에 넘김
- ② [반복]DMA 제어기는 소스에서 목적지로 데이터를 보내도록 장치제어기에 요청
- ③ 원하는 양의 입출력이 끝나면 DMA 제어기는 인터럽트 제어기에 신호를 보내어
- ④ 인터럽트를 발생시켜 CPU에게 입출력 작업이 모두 끝났음을 알림



입출력 처리 유형

■ DMA 방법 – 문제 및 해결

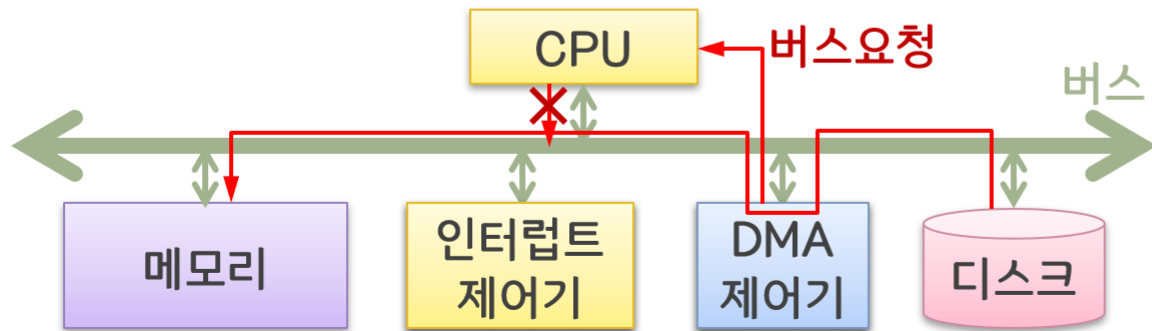
- CPU와 DMA 제어기가 동시에 메모리 액세스를 시도하면 충돌 발생
- 사이클 스틸링(cycle stealing)
 - » CPU보다 DMA 제어기에 우선권 부여
 - » 즉, CPU로부터 메모리 사이클을 훔쳐 내는 것



입출력 처리 유형

■ DMA 방법 – 문제 및 해결

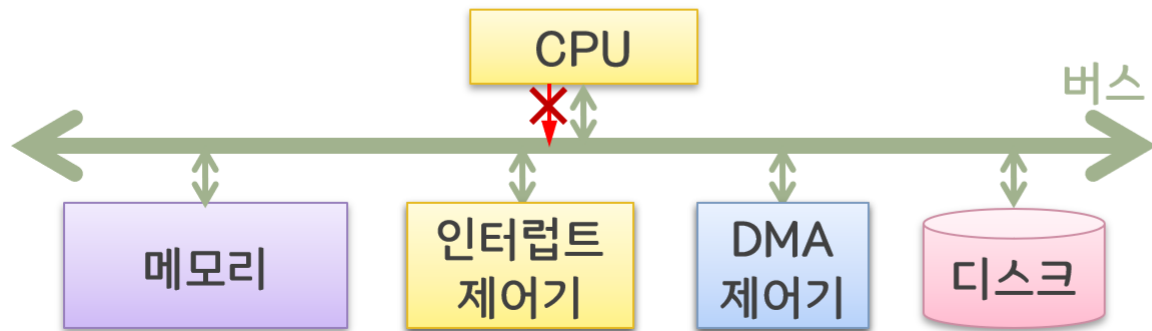
- CPU와 DMA 제어기가 동시에 메모리 액세스를 시도하면 충돌 발생
- 사이클 스틸링(cycle stealing)
 - » CPU보다 DMA 제어기에 우선권 부여
 - » 즉, CPU로부터 메모리 사이클을 훔쳐 내는 것



입출력 처리 유형

■ DMA 방법 – 문제 및 해결

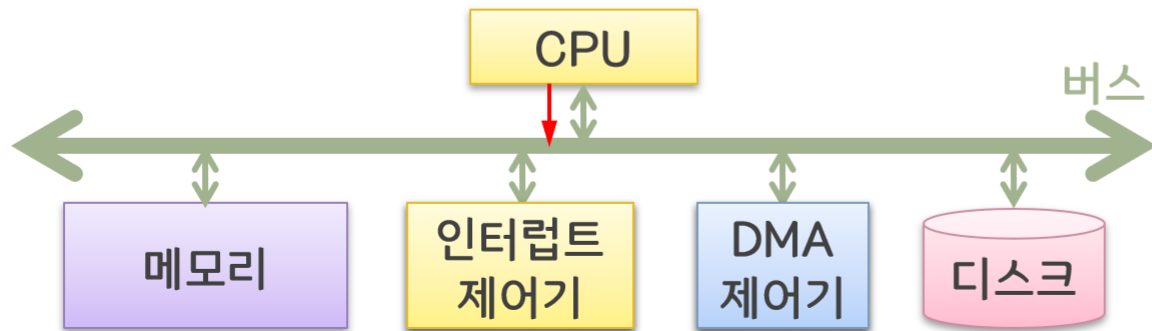
- CPU와 DMA 제어기가 동시에 메모리 액세스를 시도하면 충돌 발생
- 사이클 스틸링(cycle stealing)
 - » CPU보다 DMA 제어기에 우선권 부여
 - » 즉, CPU로부터 메모리 사이클을 훔쳐 내는 것

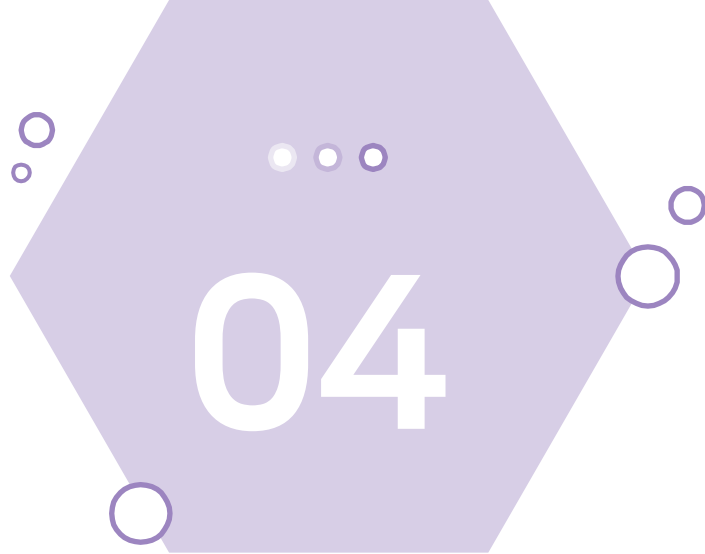


입출력 처리 유형

■ DMA 방법 – 문제 및 해결

- CPU와 DMA 제어기가 동시에 메모리 액세스를 시도하면 충돌 발생
- 사이클 스틸링(cycle stealing)
 - » CPU보다 DMA 제어기에 우선권 부여
 - » 즉, CPU로부터 메모리 사이클을 훔쳐 내는 것





입출력 관리

입출력 관리

버퍼링

- CPU의 데이터 처리 속도와 입출력장치의 데이터 전송 속도의 차이로 인한 문제를 버퍼를 통해 해결

- 세 가지 방법

» 단일 버퍼링, 이중 버퍼링, 순환 버퍼링

→ 비효율적: 저장과 처리가 동시에 이루어지지 못함

★ 버퍼(buffer)

입출력 데이터 등의 정보를 전송할 때 일시적 데이터 저장 장소로 사용되는 메모리의 일부



입출력 관리

버퍼링

- CPU의 데이터 처리 속도와 입출력장치의 데이터 전송 속도의 차이로 인한 문제를 버퍼를 통해 해결
- 세 가지 방법
 - » 단일 버퍼링, 이중 버퍼링, 순환 버퍼링

★ 버퍼(buffer)

입출력 데이터 등의 정보를 전송할 때 일시적 데이터 저장 장소로 사용되는 메모리의 일부



입출력 관리

버퍼링

- CPU의 데이터 처리 속도와 입출력장치의 데이터 전송 속도의 차이로 인한 문제를 버퍼를 통해 해결
- 세 가지 방법
 - » 단일 버퍼링, 이중 버퍼링, 순환 버퍼링

★ 버퍼(buffer)

입출력 데이터 등의 정보를 전송할 때 일시적 데이터 저장 장소로 사용되는 메모리의 일부



입출력 관리

버퍼링

- CPU의 데이터 처리 속도와 입출력장치의 데이터 전송 속도의 차이로 인한 문제를 버퍼를 통해 해결
- 세 가지 방법
 - » 단일 버퍼링, 이중 버퍼링, 순환 버퍼링

★ 버퍼(buffer)

입출력 데이터 등의 정보를 전송할 때 일시적 데이터 저장 장소로 사용되는 메모리의 일부



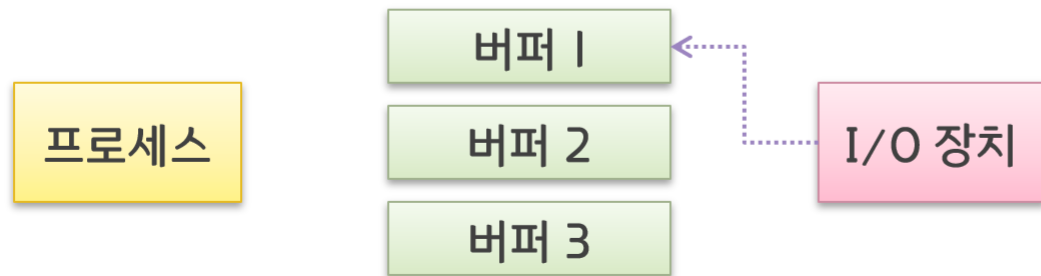
입출력 관리

버퍼링

- CPU의 데이터 처리 속도와 입출력장치의 데이터 전송 속도의 차이로 인한 문제를 버퍼를 통해 해결
- 세 가지 방법
 - » 단일 버퍼링, 이중 버퍼링, 순환 버퍼링

★ 버퍼(buffer)

입출력 데이터 등의 정보를 전송할 때 일시적 데이터 저장 장소로 사용되는 메모리의 일부



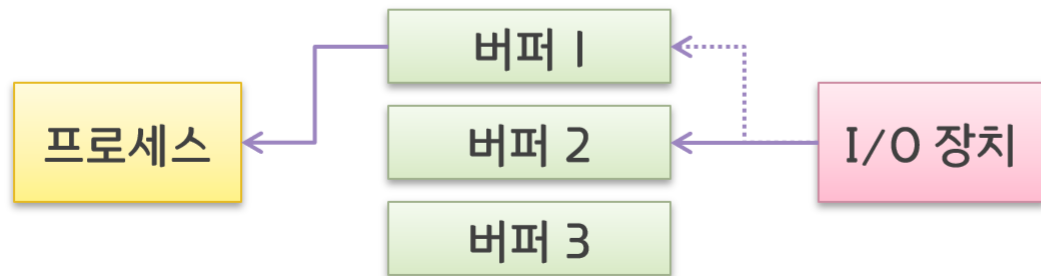
입출력 관리

버퍼링

- CPU의 데이터 처리 속도와 입출력장치의 데이터 전송 속도의 차이로 인한 문제를 버퍼를 통해 해결
- 세 가지 방법
 - » 단일 버퍼링, 이중 버퍼링, 순환 버퍼링

★ 버퍼(buffer)

입출력 데이터 등의 정보를 전송할 때 일시적 데이터 저장 장소로 사용되는 메모리의 일부



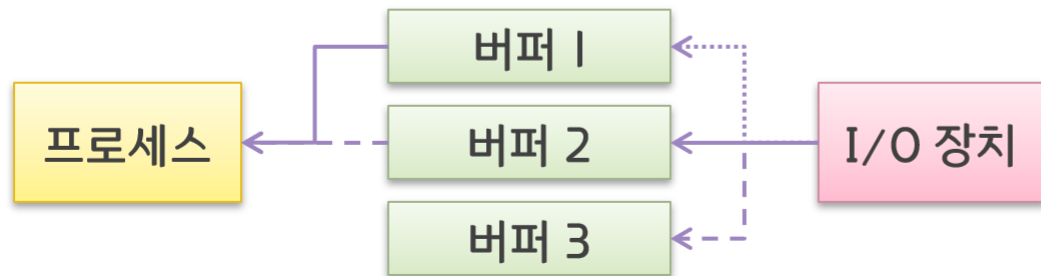
입출력 관리

버퍼링

- CPU의 데이터 처리 속도와 입출력장치의 데이터 전송 속도의 차이로 인한 문제를 버퍼를 통해 해결
- 세 가지 방법
 - » 단일 버퍼링, 이중 버퍼링, 순환 버퍼링

★ 버퍼(buffer)

입출력 데이터 등의 정보를 전송할 때 일시적 데이터 저장 장소로 사용되는 메모리의 일부



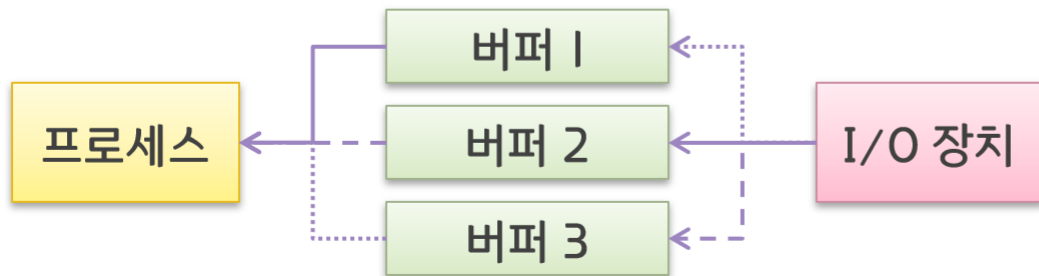
입출력 관리

버퍼링

- CPU의 데이터 처리 속도와 입출력장치의 데이터 전송 속도의 차이로 인한 문제를 버퍼를 통해 해결
- 세 가지 방법
 - » 단일 버퍼링, 이중 버퍼링, 순환 버퍼링

★ 버퍼(buffer)

입출력 데이터 등의 정보를 전송할 때 일시적 데이터 저장 장소로 사용되는 메모리의 일부



입출력 관리

버퍼링

- CPU의 데이터 처리 속도와 입출력장치의 데이터 전송 속도의 차이로 인한 문제를 버퍼를 통해 해결
- 세 가지 방법
 - » 단일 버퍼링, 이중 버퍼링, 순환 버퍼링

★ 버퍼(buffer)

입출력 데이터 등의 정보를 전송할 때 일시적 데이터 저장 장소로 사용되는 메모리의 일부



입출력 관리

스폴링

- 입출력 프로세스와 저속 입출력장치 사이의 데이터 전송을 자기 디스크와 같은 고속 장치를 통하도록 하는 것(일종의 버퍼링)
- 입출력 작업이 빨리 끝나게 할 수 있음
- 전용장치를 가상장치로 변화시켜 줌





강의를 마쳤습니다.

다음시간에는

12강. 저장장치 및 파일