

본 강의에서 수업자료로 이용되는 저작물은
저작권법 제25조 수업목적 저작물 이용 보상금제도에 의거,
한국복제전송저작권협회와 약정을 체결하고 적법하게 이용하고 있습니다.
약정범위를 초과하는 사용은 저작권법에 저촉될 수 있으므로
수업자료의 재 복제, 대중 공개·공유 및 수업 목적 외의 사용을 금지합니다.

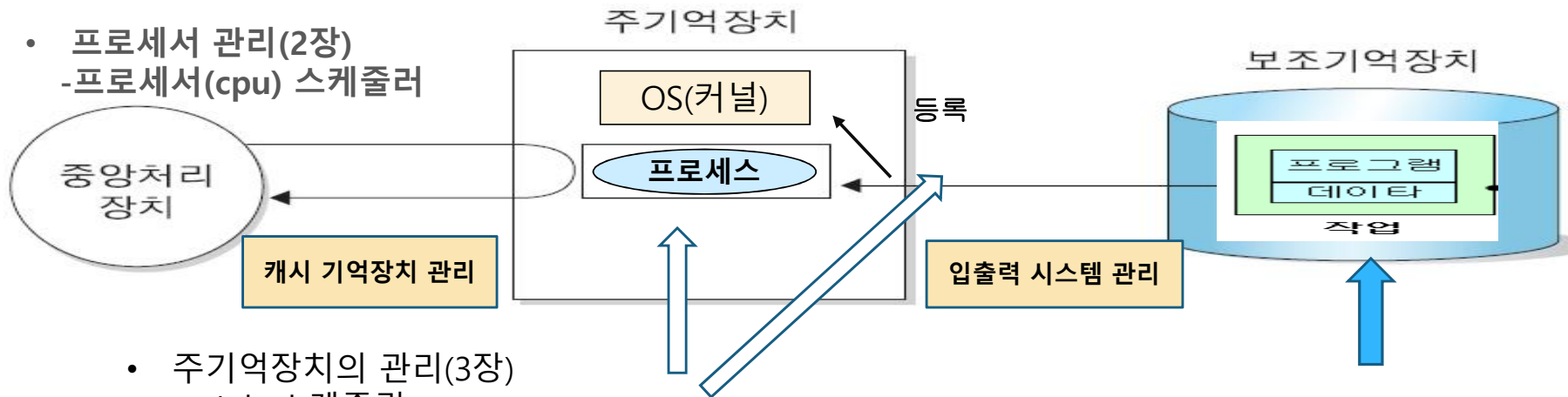
2023. 3 . 2 .

부천대학교·한국복제전송저작권협회

운 영 체 제

입출력 시스템 관리

5장 디스크 스케줄링과 파일 시스템



- 주기억장치의 관리(3장)
 - Job 스케줄링
 - 주소 바인딩
 - 주기억장치 관리 기법
 - => 인출 기법, 배치 기법, 교체 기법, 할당 기법(연속)
- 가상 메모리 관리(4장)
 - 분산 할당 기법
 - => 페이징 기법, 세그먼테이션 기법
 - 페이지 교체 기법
 - => FIFO(First In First Out),
OPT(Optimal Replacement), LRU, LFU, NUR, SCR

- 디스크 스케줄링 (5장)
 - FCFS, SSTF, SCAN, C-SCAN
 - => 탐색시간 최소화
- 파일 시스템(5장)
 - 파일 구조
 - 디렉토리 구조

학습 내용(1)

입출력 시스템 관리

- 입출력 시스템과 입출력 모듈의 개념
- 입출력 모듈의 구성
- 입출력 장치의 연결과 데이터 전송
- 입출력 모듈의 기능
- 입출력 제어 방법
- 커널 입출력 서브 시스템

입출력 관리 개념

■ 입출력 관리

- 프로그램의 실행 속도는 입출력 시스템의 성능에 영향을 많이 받는다.
- 운영체제의 역할 중 입출력을 관리하고 제어하는 일은 매우 중요하다.
- 입출력장치는 중앙처리장치와 주기억장치에 비하여 동작 속도가 현저하게 느려서 직접적으로 컴퓨터에 연결되지 않고 중간에 별도의 장치를 필요로 한다.
- 입출력 모듈을 통해 상호 연결되어 있다

■ 관련 개념 및 용어

- 입출력 시스템
- 입출력 장치
- 입출력 모듈
 - 입출력 채널
 - 입출력 인터페이스
 - 입출력 프로세서
 - 입출력 제어기
 - 장치 제어기

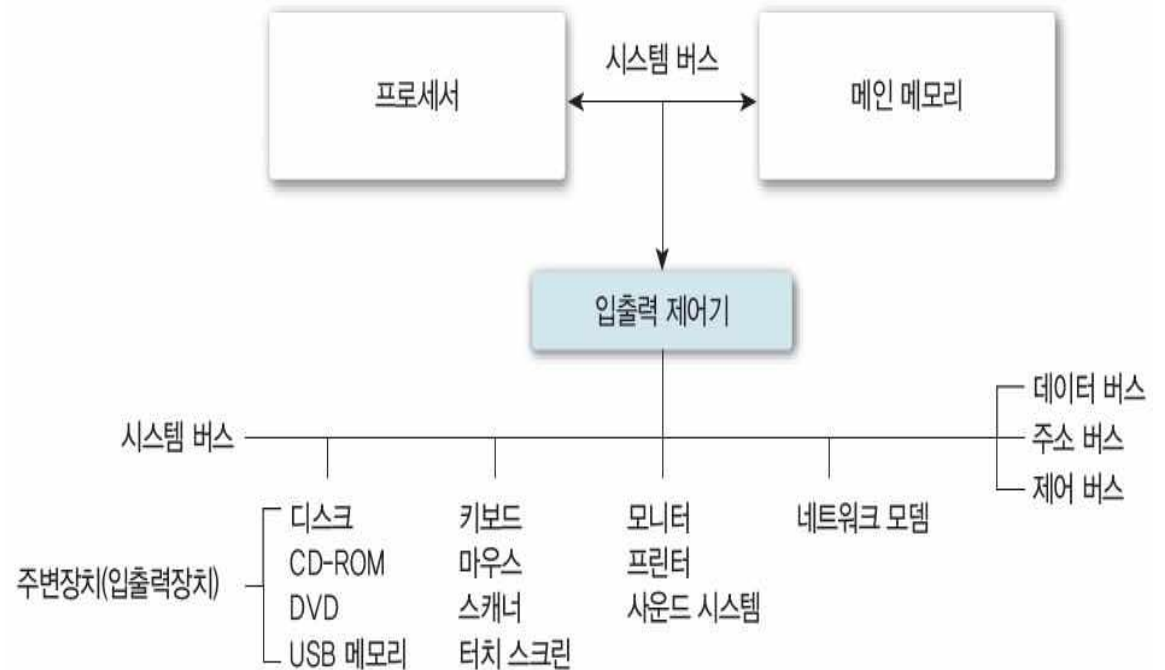
=> 처리하는 일의 정도에 따라 다양한 이름으로 불림

• 입출력 모듈의 기능

- » 중앙처리장치(프로세서)와의 통신
- » 입출력장치들과의 통신
- » 입출력장치의 제어(control)와 타이밍(timing) 조정
- » 입출력장치의 주소 지정
- » 데이터 버퍼링(data buffering) 기능을 수행
- » 오류 검출(error detection)

• 입출력의 제어 방법

- » 프로세서 제어 입출력(프로그램 제어 입출력, 인터럽트 기반 입출력)
- » DMA 입출력
- » 입출력 채널

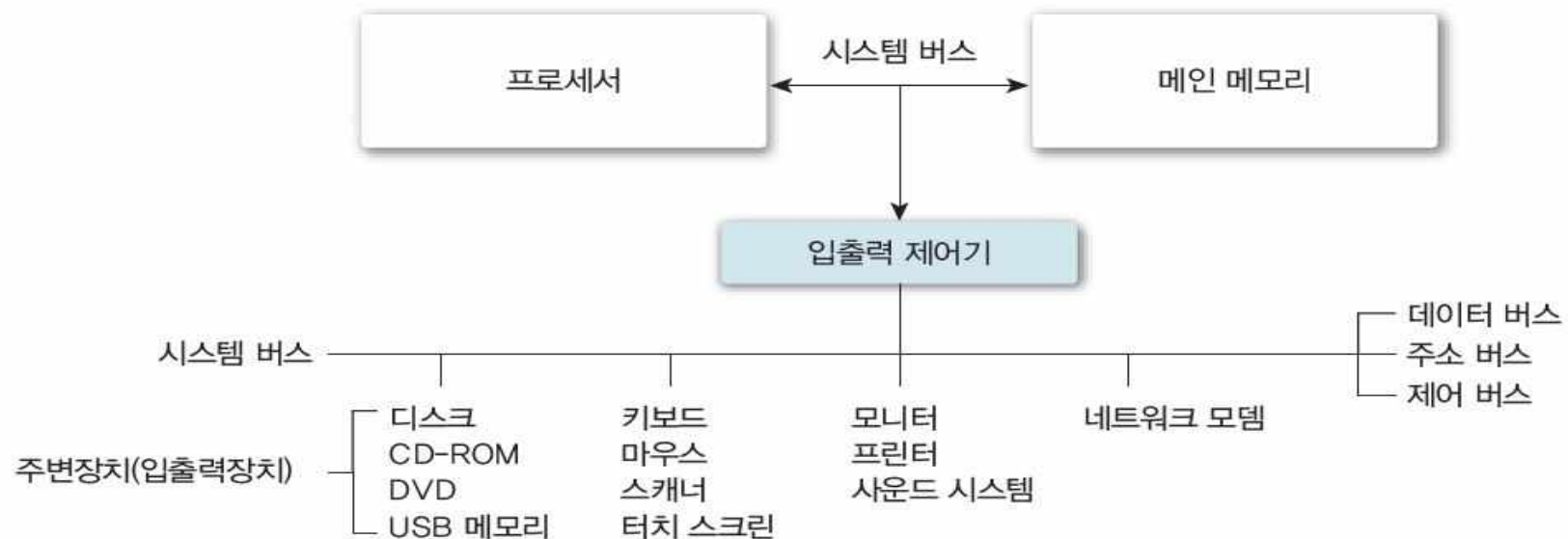


컴퓨터 하드웨어 구성과 입출력 모듈

입출력 시스템과 입출력 모듈의 개념

■ 입출력 시스템과 입출력 모듈의 개념

- 입출력 시스템은 모니터나 프린터 같은 하드웨어 장치 뿐만 아니라 입출력 모듈까지 포함
- 물리적 입출력장치가 실제로 입출력을 수행하고, **입출력 모듈**은 메모리나 프로세서, 레지스터 등 내부 저장장치와 물리적 입출력 장치 사이의 이진 정보를 전송 방법 제공
- **입출력 모듈**이 프로세서를 대신하여 입출력과 관련된 복잡한 일을 처리하면 입출력 채널 또는 입출력 프로세서가 되고, 단순히 프로세서의 입출력과 관련된 일을 담당하면 입출력 제어기 또는 장치 제어기가 됨

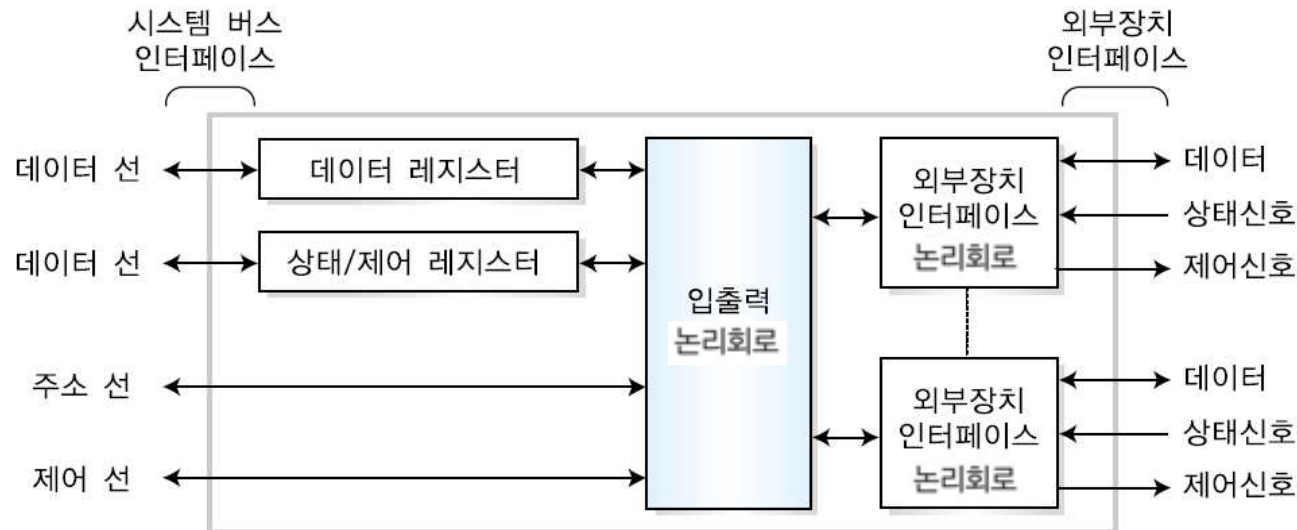


컴퓨터 하드웨어 구성과 입출력 모듈

입출력 모듈의 구성

■ 입출력 모듈의 구성

- 입출력장치의 수와 구조에 따라 다양하게 구성 가능
- 일반적인 구성



- » 데이터 레지스터 : 입출력 모듈로 들어가거나 나오는 데이터는 데이터 레지스터에 일시 저장, 버퍼링 기능을 위해서 일시적으로 저장된다.
- » 상태/제어 레지스터 : 현재의 상태와 오류를 저장하기 위한 레지스터로, 중앙처리장치에서 보낸 제어 정보를 저장하기 위한 제어 레지스터로도 동작한다.
- » 입출력 논리회로는 상태 레지스터가 제어하는 장치의 주소를 인식하고 주소도 발생 할 수 있어야 하므로 연결된 각 장치와 인터페이스를 제어하려고 장치 인터페이스를 포함한다. 그리고 제어 버스로 프로세서와 교신하는 회로
- » 제어 line들 : 중앙처리장치가 입출력 모듈로 명령을 보내는데 사용한다.
- » 주소 line들 : 중앙처리장치는 주소선을 통해서 입력된 여러 입출 모듈의 주소들 중에서 자신만의 주소를 인식, 연결된 입출력장치들의 주소도 알 수 있어야 한다.
- » 연결된 입출력장치를 제어하기 위한 데이터, 상태 신호, 제어 신호를 가지고 있다.

- 입출력 모듈은 연결된 입출력장치를 제어하는데 필요한 세부적인 사항들을 모두 처리해주기 때문에 중앙처리장치의 부담을 덜어준다.

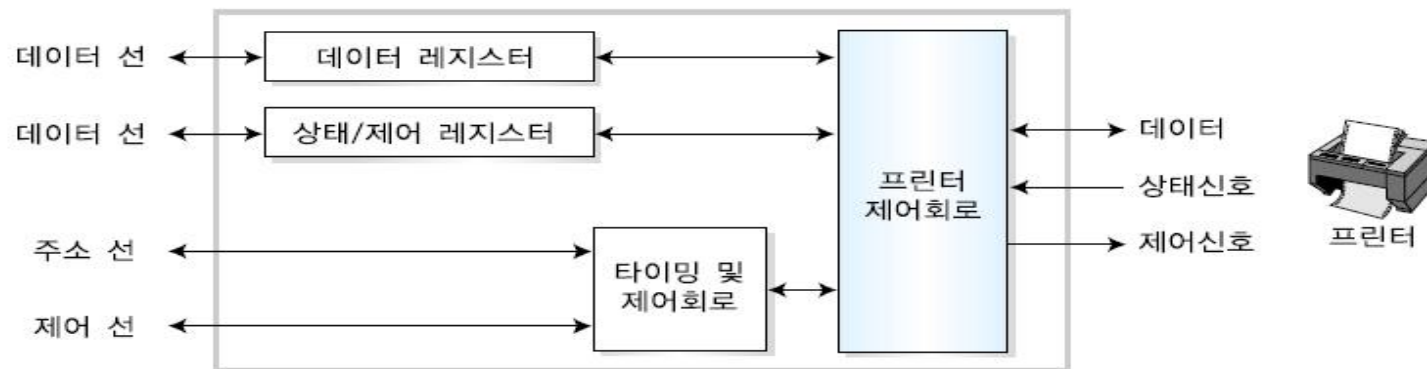
입출력 모듈의 구성

■ 입출력 모듈의 구성 예

■ 프린터 모듈의 동작

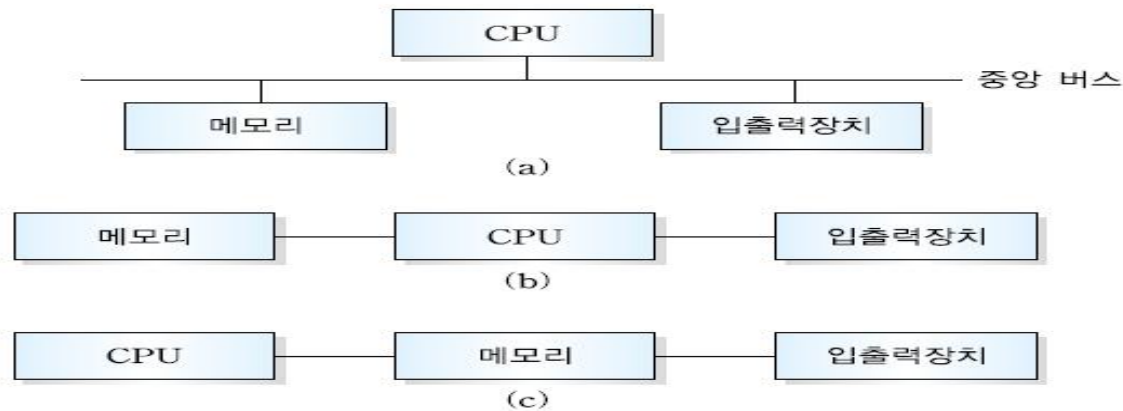
■ 중앙처리장치가 프린터 장치에 데이터를 전송하는 과정

- 1단계 : 중앙처리장치가 프린터 입출력 모듈에게 프린터의 상태 검사 요청한다.
- 2단계 : 입출력 모듈은 상태/제어 레지스터를 통해서 요청을 수신하고, 프린터 제어회로를 통해서 프린터의 상태를 검사한다.
- 3단계 : 프린터 상태 검사를 통해 프린트할 준비가 되었는지, 혹은 다른 데이터를 프린트하는 중인지의 결과를 상태 신호로 중앙처리장치로 전달한다.
- 4단계 : 프린터가 데이터 수신 상태면, 중앙처리장치는 입출력 모듈에 출력 명령과 데이터를 전송한다. 만약 다른 데이터를 프린트하는 중이라면 대기한다.
- 5단계 : 입출력 모듈은 프린트 동작을 수행하기 위한 제어 신호와 함께 출력될 데이터를 프린터로 전송한다.



입출력 장치의 연결과 데이터 전송

- 컴퓨터와 입출력장치가 연결되는 방법에 따라 컴퓨터 동작 특성이 다르다.
- 컴퓨터 시스템 구성방법



- (a) 중앙 버스에 중앙처리장치, 주기억장치 그리고 입출력장치가 연결된 형태다.
 - » 입출력장치는 입출력 모듈을 통해서 연결된다.
 - » 이 연결은 제어 동작에 의해서 각 장치가 독립적으로 원활한 데이터 전송이 가능하다.
- (b) CPU가 중앙이고 좌우에는 주기억장치와 입출력장치가 연결되는 형태다.
 - » 주기억장치에서 외부 입출력장치로 직접 데이터를 전송할 수 없고 CPU에 의해서 전송이 결정
- (c) 중앙처리장치, 주기억장치, 입출력 장치 순의 직렬 연결 형태다.
 - » 입출력장치가 직접으로 CPU에 데이터를 전송할 수 없고 주기억장치를 꼭 경유해야만 한다.

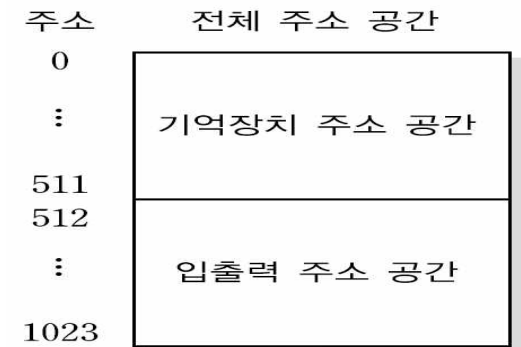
입출력 장치의 연결과 데이터 전송

■ 입출력장치의 주소지정

- 입출력 모듈에는 여러 개의 입출력장치가 연결되어 있으므로, 각 장치들을 구분할 수 있도록 고유의 번호 또는 주소가 지정되어 있다.
- 기억장치-사상 방식과 분리형 입출력 방식의 두 가지 주소 지정 방식이 있다.

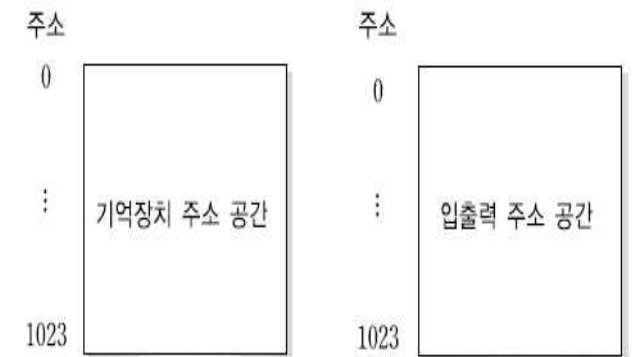
■ 기억장치-사상 방식 (memory-mapped)

- 입출력장치와 주기억장치는 하나의 주소 공간을 공유한다.
 - 기억장치 주소 영역의 일부분을 입출력장치의 주소 영역으로 할당하는 방식이다.
 - 기억장치의 읽기/쓰기 신호를 입출력장치의 읽기/쓰기 신호로 사용이 가능하다.
 - 프로그램에서 기억장치 관련 명령어들을 입출력장치 제어에도 사용이 가능하다.
 - 입출력장치가 기억장치 주소 영역을 사용하므로 기억장치의 주소 공간이 감소한다.
- 기억장치-사상 방식의 개념
- 10비트의 주소영역을 사용한다고 할 때 0번지~511번지까지의 상위 512개 주소는 기억장치의 주소 공간을 위해서 할당한다.
 - 나머지 512번지~1023번지까지의 하위 512개 주소는 입출력장치들의 주소 공간을 위해서 할당한다.



■ 분리형 입출력 방식(isolated I/O 또는 I/O mapped)

- 입출력장치의 주소 공간을 기억장치 주소 공간과는 별도의 기억장치에 할당하는 방식이다.
 - 입출력 제어를 위해서 별도의 입출력 명령어를 사용하기 때문에, 별도의 입출력장치에 대한 읽기 쓰기 신호가 필요하다.
 - 입출력 주소 공간은 기억장치 주소 공간과는 별도로 지정이 가능하지만, 입출력 제어를 위해 입출력장치 명령어들만 이용할 수 있기 때문에 프로그래밍이 복잡해져서 불편하다.
- 분리형 입출력 방식의 개념
- 주소 비트가 각각 10비트일 때 기억장치 주소와 입출력 주소는 각각 1024개씩 할당이 가능하다.



입출력 장치의 연결과 데이터 전송

■ 입출력 데이터 전송

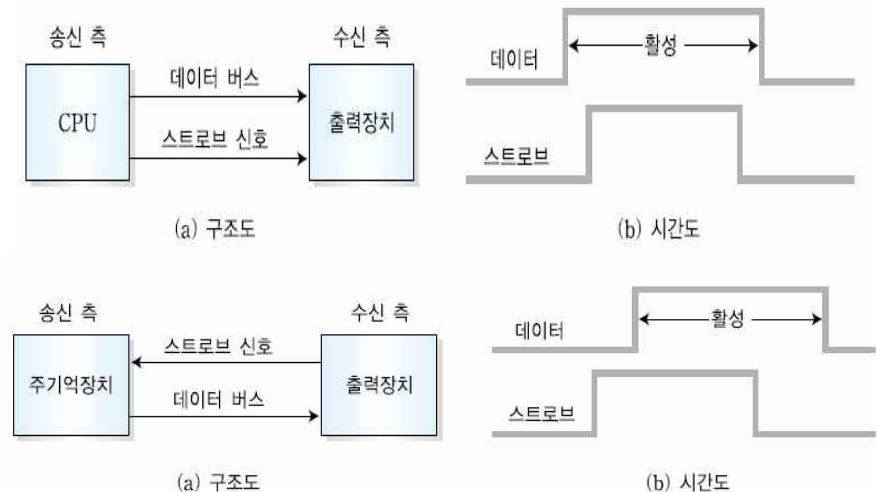
- 독립된 2개 이상의 입력장치 및 출력장치가 비동기적으로 데이터를 전송하는 경우에는 데이터의 전송을 알리는 방법이 필요하다.
- **데이터 전송을 알리는 방법**
 - **스트로브 신호를 이용하는 방법**
 - **제어 신호를 이용하는 핸드셰이킹 방법**

■ 스트로브 신호(strobe)

- 송신 측에서 데이터를 전송하는 경우 전송되는 것을 수신 측에 알려주기 위해 별도의 신호를 스트로브(Strobe) 신호라 한다.
- 이 신호를 전달하기 위해서는 별도의 회선이 필요하므로 데이터 버스 외에 추가적인 회선을 설치해야 한다.
- 스트로브 신호를 보내는 방법은 송신 측에서 수신 측으로 보내는 방법과 수신 측에서 송신 측으로 보내는 2가지 방법이 존재한다.

■ 스트로브 신호 전송 방법

- 송신 측에서 수신 측으로 스트로브 신호를 보내는 방법
 - 송신 측의 CPU에서 데이터를 데이터버스에 실어 데이터 펄스 신호를 안정적으로
 - 활성화 한 후 스트로브 신호를 수신 측에 해당하는 출력장치에 보낸다.
 - 출력장치는 스트로브 신호와 데이터 버스에서 데이터를 수신한다.
- 수신 측에서 송신 측으로 스트로브 신호를 보내는 방법
 - 수신 측에서 스트로브 신호를 송신 측에 전달하여 데이터에 대한 전송을 요청한다.
 - 송신 측에서는 데이터 버스에 전송할 데이터를 보내고 수신 측에서는 이 데이터를 수신하게 된다.



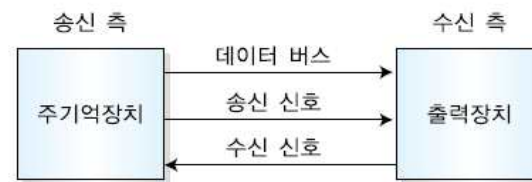
입출력 장치의 연결과 데이터 전송

■ 입출력 데이터 전송

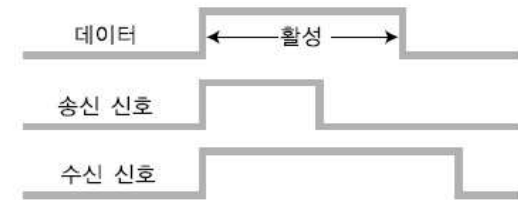
- 독립된 2개 이상의 입력장치 및 출력장치가 비동기적으로 데이터를 전송하는 경우에는 데이터의 전송을 알리는 방법이 필요하다.

- **데이터 전송을 알리는 방법**

- **스트로브 신호를 이용하는 방법**
- **제어 신호를 이용하는 핸드셰이킹 방법**



(a) 구조도



(b) 시간도

■ 핸드셰이킹(handshaking)

- 송수신 측 양쪽에서 제어 신호를 보내서 데이터의 전송을 알려주는 방법이다.
- 데이터 버스 외에 양쪽에서 제어 신호를 보내주는 별도의 회선을 각각 가지고 있어야 한다.
- 핸드셰이킹 방법의 구조와 시간 펄스
 - 데이터 버스 외에 제어 버스를 통해서 제어 신호를 송신하는 것을 확인할 수 있다.
 - 시간도에서 데이터 전송이 시작되면 주기억장치는 송신 제어 신호를 전송하고 출력장치는 송신 신호를 확인하고 데이터를 수신한다.
 - 수신이 완료되면 출력장치는 주기억장치에 수신 제어 신호를 전송하여 송신 측에 수신이 완료된 것을 알려준다.

■ 핸드셰이킹 방법 특징

- 송신 측과 수신 측이 동시에 동작하는 방식으로 어느 한쪽의 장치가 잘못되면 데이터의 전송이 이루어지지 않기 때문에 높은 신뢰성을 갖는다.
- 또한 이 방법은 하나의 송신 장치에서 여러 개의 수신 장치에 데이터를 전송할 수도 있다.

입출력 모듈의 기능

■ 입출력 모듈의 기능

- 중앙처리장치(프로세서)와의 통신
 - 입출력 모듈은 또한 입출력장치간 통신이 가능해야 함
 - 입출력 모듈은 명령 해독 → 데이터 교환 → 상태 보고 → 주소 인식 과정 거쳐 메시지 인식
- 입출력장치들과의 통신
- 입출력장치의 제어(control)와 타이밍(timing) 조정
 - 내부 자원과 데이터 입출력 등 다양한 동작을 제어하고 타이밍 기능을 제공 입출력 모듈이 외부장치의 타이밍과 데이터 형식, 기계적인 세부 사항들을 처리하므로 프로세서는 단순히 파일 열기나 닫기 명령만으로도 장치를 제어 가능
- 입출력장치의 주소 지정
 - 프로세서가 입출력장치에 명령을 보내려면 입출력장치의 식별자를 주소로 지정 해야 함
 - 주소를 지정하는 방법
 - 메모리 주소 공간의 일부를 입출력 주소 공간으로 공유하는 메모리 매핑 주소 방법: 기억장치-사상 방식 (memory-mapped)
 - » 메모리 매핑 주소 방법은 임의의 기계 명령과 주소를 입출력장치를 지정하는 데 사용하는 등 다양한 유연성, 데이터 전송 완료를 통보하려고 인터럽트 신호 생성
 - 메모리 주소 공간과 입출력 주소 공간이 별도로 있는 전용 입출력 주소 방법: 분리형 입출력 방식(isolated I/O 또는 I/O mapped)
- 데이터 버퍼링(data buffering) 기능을 수행
 - 버퍼링을 이용하여 전송속도 조절
- 오류 검출

입출력 모듈의 기능

■ 중앙처리장치와의 통신

■ 중앙처리장치와 외부 장치 간의 통신을 수행하기 위한 입출력 모듈의 기능

- 명령 해석(Command Decoding)
 - » 입출력 모듈은 CPU에서 받은 명령을 해석하고, 제어 버스를 통해서 제어 신호로 명령을 보낸다.
 - » CPU에서 하드디스크나 CD-ROM으로 전달되는 명령은 데이터의 저장이나 인출을 위한 것으로 READ SECTOR, WRITE SECTOR, SEEK track number, SCAN record ID 와 같은 명령들이다.
- 데이터(Data) 교환
 - » CPU 입출력 모듈의 가장 근본적인 기능으로 데이터 교환은 데이터 버스를 통하여 이루어 진다.
- 상태 보고(Status Reporting)
 - » 주변장치들이 저속으로 동작하기 때문에 입출력 모듈의 상태를 확인하는 것은 중요하다.
 - » 입출력 모듈은 상태를 확인하여 BUSY, READY, 결함 상태 등의 상태 보고를 수행한다.
- 주소 인식(Address Recognition)
 - » 여러 종류의 입출력장치들을 구별하기 위해서는 주소가 필요하다. 따라서 입출력 모듈은 제어하는 여러 주변장치의 주소를 인식하고 있어야 한다.

■ 입출력 모듈은 또한 입출력장치간 통신이 가능해야 함

- 입출력 장치 간 통신에서도 명령들과 상태 정보 및 데이터가 포함된다

입출력 모듈의 기능

■ 입출력장치의 제어와 타이밍 조정

- 내부 장치들과 외부 장치들 사이의 데이터 흐름을 조정하기 위한 기능
- 외부 장치에서 중앙처리장치로 데이터가 전송되는 것을 제어하는 순서
 - 1단계 : 중앙처리장치가 입출력 모듈에게 입출력장치의 상태를 검사하도록 요청한다
 - 2단계 : 입출력 모듈이 상태를 보고한다
 - 3단계 : 만약 입출력장치가 준비 상태라면, 중앙처리장치가 데이터 전송을 요청한다
 - 4단계 : 입출력 모듈이 입출력장치로부터 데이터를 수신한다.
 - 5단계 : 입출력 모듈이 중앙처리장치로 데이터를 보내준다
- 위의 5단계 과정에서 입출력 모듈은 입출력장치에 대하여 제어 기능을 수행하고 적절한 시기에 데이터를 전송할 수 있도록 타이밍 조정기능을 수행함

입출력 모듈의 기능

■ 데이터 버퍼링과 오류검출

■ 데이터 버퍼링(Data Buffering)

- 컴퓨터 내부에서 입출력 모듈로 전달되는 데이터의 전송 속도는 고속이다. 그리고 이렇게 전달된 데이터는 입출력 모듈 내의 버퍼에 일시적으로 저장되었다가 적절한 전송 속도로 주변장치로 보내지는 버퍼링 기능을 수행한다.
- 입출력장치에서 컴퓨터 내부로 전달되는 저속의 데이터는 주기억장치 또는 중앙처리장치의 동작에 영향을 주지 않도록 입출력 모듈의 버퍼에 고속의 데이터 전송률 될 수 있도록 저장되었다가 전송된다.
- 입출력 모듈은 저속의 전송률과 고속의 전송률에 모두 동작 할 수 있어야 한다.

■ 오류 검출

- 입출력장치들의 오류를 검사하고 오류가 발생하면 중앙처리장치로 보고 할 수 있어야 한다.
- 입출력장치에서 발생하는 오류는 기계 및 전기적 오류와 데이터 전송 중에 발생하는 비트 오류 등이 있다.
- 기계 및 전기적 오류에는 프린터의 종이 걸림, 하드디스크의 불량 디스크 트랙 등이 대표적이다. 그리고 전송오류를 검출하는데 사용되는 오류-검출 코드는 일반적으로 패리티 비트를 사용한다.

학습 내용(2)

입출력 시스템 관리

- 입출력 시스템과 입출력 모듈의 개념
- 입출력 모듈의 구성
- 입출력 장치의 연결과 데이터 전송
- 입출력 모듈의 기능
- **입출력 제어 방법**
 - **프로세서 제어 입출력** : 중앙처리장치가 직접 입출력장치를 제어하는 방식
 - **DMA 입출력** : DMA(Direct Memory Access)로 직접 기억장치 액세스 입출력 제어 방식
 - **입출력 채널** : 채널 제어기 또는 입출력 전용 프로세서에 의한 입출력 제어 방식
- **커널 입출력 서브 시스템**
 - **커널이 제공하는 입출력 관련 서비스**

입출력 관리 개념

■ 입출력 관리

- 프로그램의 실행 속도는 입출력 시스템의 성능에 영향을 많이 받는다.
- 운영체제의 역할 중 입출력을 관리하고 제어하는 일은 매우 중요하다.
- 입출력장치는 중앙처리장치와 주기억장치에 비하여 동작 속도가 현저하게 느려서 직접적으로 컴퓨터에 연결되지 않고 중간에 별도의 장치를 필요로 한다.
- 입출력 모듈을 통해 상호 연결되어 있다

■ 관련 개념 및 용어

- 입출력 시스템
- 입출력 장치
- 입출력 모듈
 - 입출력 채널
 - 입출력 인터페이스
 - 입출력 프로세서
 - 입출력 제어기
 - 장치 제어기

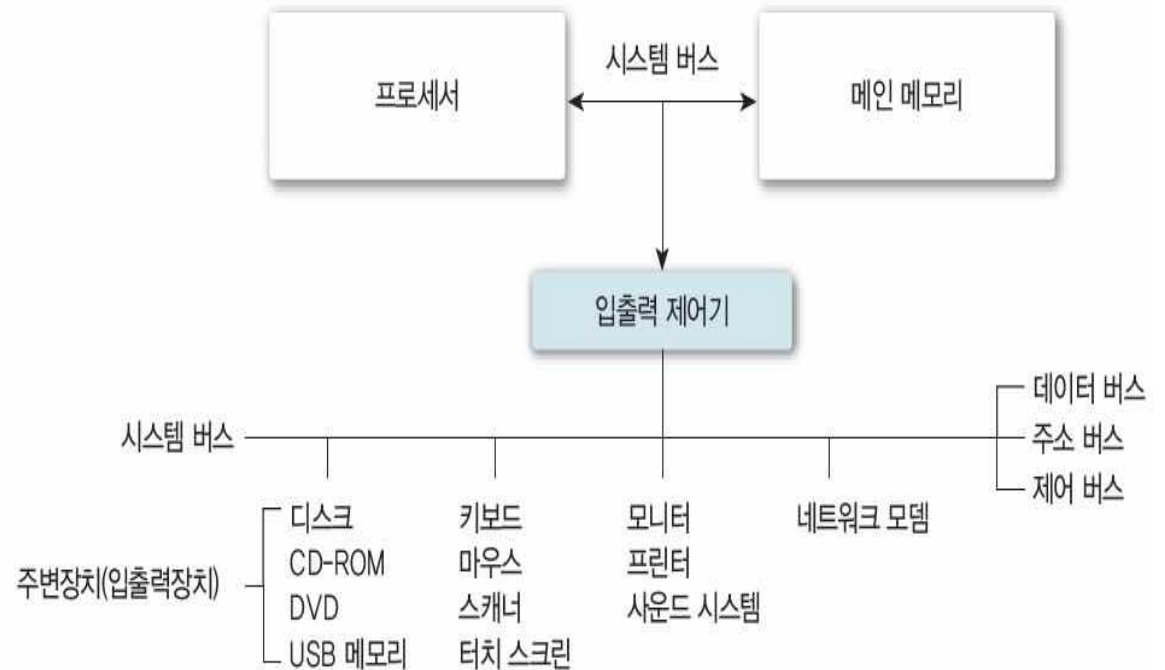
=> 처리하는 일의 정도에 따라 다양한 이름으로 불림

• 입출력 모듈의 기능

- » 중앙처리장치(프로세서)와의 통신
- » 입출력장치들과의 통신
- » 입출력장치의 제어(control)와 타이밍(timing) 조정
- » 입출력장치의 주소 지정
- » 데이터 버퍼링(data buffering) 기능을 수행
- » 오류 검출(error detection)

• 입출력의 제어 방법

- » 프로세서 제어 입출력(프로그램 제어 입출력, 인터럽트 기반 입출력)
- » DMA 입출력
- » 입출력 채널



컴퓨터 하드웨어 구성과 입출력 모듈

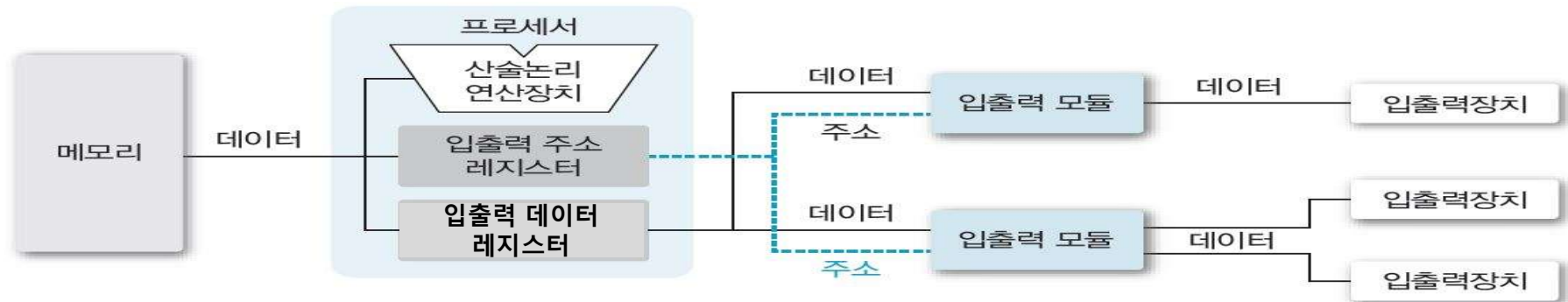
입출력 제어 방법

- 입출력장치가 컴퓨터의 내부 장치와 원활한 통신을 수행하려면 통신을 제어할 수 있는 제어 기법이 필요
- 입출력 제어 방법
 - => 프로세서 역할에 따라
 - 프로세서 제어 입출력 : 중앙처리장치가 직접 입출력장치를 제어하는 방식
 - 입력장치 및 출력장치를 직접 제어하는 방법으로, 데이터 전송뿐만 아니라 데이터 상태 검사 등의 모든 것을 중앙처리장치가 직접 명령을 수행한다
 - 프로그램 제어 입출력 : 프로그램에 의한 입출력 제어 방식(폴링(Polling) 방식)
 - 인터럽트 기반 입출력 : 인터럽트에 의한 입출력 제어 방식(인터럽트(Interrupt-driven I/O) 방식)
 - DMA 입출력 : DMA(Direct Memory Access)로 직접 기억장치 액세스 입출력 제어 방식
 - 프로세서 개입 없이 직접 주기억 장치와 DMA 사이에서 일련의 입출력 동작이 이루어짐
 - 프로그램이 실행하는 동안 인터럽트의 수를 최소화 시킴
 - 입출력 채널 : 채널 제어기 또는 입출력 전용 프로세서에 의한 입출력 제어 방식

입출력 제어 방법

■ 프로그램 제어 입출력 방법

- 중앙처리장치가 프로그램을 수행하는 도중에 입출력과 관련된 명령을 만나면 해당 입출력 모듈에 명령을 보냄으로써 그 명령을 실행하는 방식
- 프로세서 내부의 입출력 데이터와 주소 레지스터를 입출력 모듈과 연결한 형태
- 주소 레지스터와 버스 사이에서 데이터를 직접 전송할 수 있는 가장 단순한 형태
- 데이터 입력할 때는 입출력 모듈을 거쳐 한 번에 한 워드 씩만 데이터 레지스터로 전송, 입출력 데이터 레지스터에서는 프로그램을 이용하여 산술 논리연산장치로 전송
- 데이터를 출력할 때는 산술논리연산장치에서 입출력 데이터 레지스터로 이동, 프로그램을 이용하여 입출력 모듈로 전송
- 구성

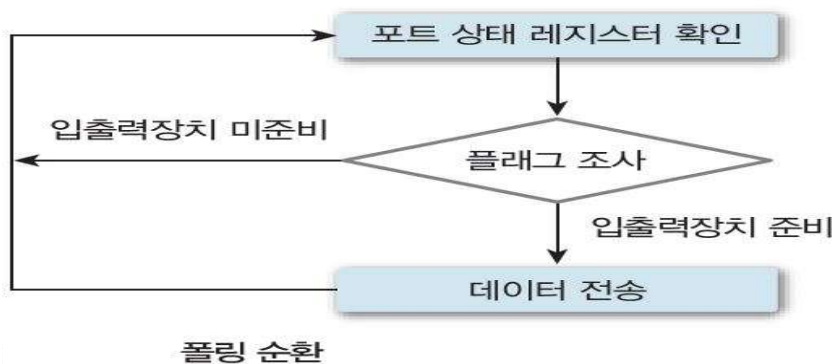


프로그램 제어 입출력 방법의 구성

입출력 제어 방법

■ 프로그램 제어 입출력 방법을 폴링polling 방법이라고도 함

- CPU와 입출력 장치 사이의 데이터 전달이 프로그램에 의해서 제어되는 방법으로 프로그램에서 CPU가 입출력 모듈의 상태 레지스터의 D와 B 플래그(flag) 상태를 검사하여 데이터를 전송 하는 방식
- CPU가 계속적으로 플래그를 검사하여 입출력 전방에 걸쳐 개입하기 때문에 비효율적이며 CPU는 다른 여산을 처리 하지 못하는 방식
- 폴링 : 상태 비트를 주기적으로 검사하여 프로세서보다 상대적으로 느린 입출력장치의 상태를 확인
- 폴링 방법으로 데이터를 전송할 때는 폴링 순환 수행
- 폴링 방법에서는 순환 횟수가 데이터 전송 성능에 영향. 너무 빈번하게 순환하면 입출력 동작의 진행 여부를 검사하는 데 시간 많이 낭비. 반대로 폴링 횟수가 너무 많으면 입출력장치가 오래 쉼



입출력 명령	내용
제어(control)	주변장치를 활성화시키고 무엇을 해야 하는지 알리는 데 사용한다.
검사(test)	입출력 모듈과 주변장치들의 상태를 검사하는 데 사용된다. 검사의 예로는 입출력 동작의 완료 여부와 오류 발생 여부를 검사하는 것이 있다.
읽기(read)	입출력 모듈이 주변장치에서 데이터를 읽고 내부 레지스터에 저장한다.
쓰기(write)	입출력 모듈이 데이터 버스에서 데이터를 받아 주변장치로 보낸다.

프로그램 입출력 방식에서 사용하는 입출력 명령들(I/O Command)

입출력 제어 방법

■ 인터럽트 기반 입출력 방법

- CPU가 계속적으로 플래그를 검사하지 않고 입출력이 필요할 때 입출력 장치에서 CPU에게 인터럽트를 요청하면 그 때 CPU가 하던 일을 멈추고 입출력 장치에 데이터를 전송하게 하는 방식
- 프로그램에 의한 입출력 방식보다 CPU의 이용 효율이 좋음.
- 입출력장치가 작업을 완료 한 후에 작업과 관련된 상태와 결과를 메모리(스택)에 저장하고 인터럽트를 발생시켜 프로세서에 알림
- 인터럽트를 받은 프로세서는 입출력 명령을 전송하고 입출력 작업 중에 다른 명령 시작
- 프로세서가 프로세서의 도움이 필요한지 입출력장치에 확인하는 대신 입출력장치가 프로세서에 직접 신호를 주는 방법
- 불규칙적이고 빠른 응답성을 요구할 때 적합
- 기본 구성



인터럽트 기반 입출력 방법의 구성

- 인터럽트(Interrupt)
 - 일시 중단이라는 의미로, 중앙처리장치가 프로그램을 실행하고 있는 도중에 다른 프로그램을 처리하기 위해 실행 중인 프로그램을 중단 상태로 만들고 다른 프로그램을 처리하는 것을 말한다.

입출력 제어 방법

■ 인터럽트 기반 입출력 방식의 읽기 동작

- 1단계 : 중앙처리장치가 입출력 모듈로 읽기(read) 명령을 보낸다.
- 2단계 : 중앙처리장치는 입출력 모듈에게 입출력 동작의 개시를 지시한다.
- 3단계 : 입출력 모듈은 주변장치에서 데이터를 읽는다. 이 과정 동안 중앙처리장치는 다른 일을 수행한다.
- 4단계 : 입출력 모듈이 중앙처리장치로 입력 완료 인터럽트 신호를 보낸다.

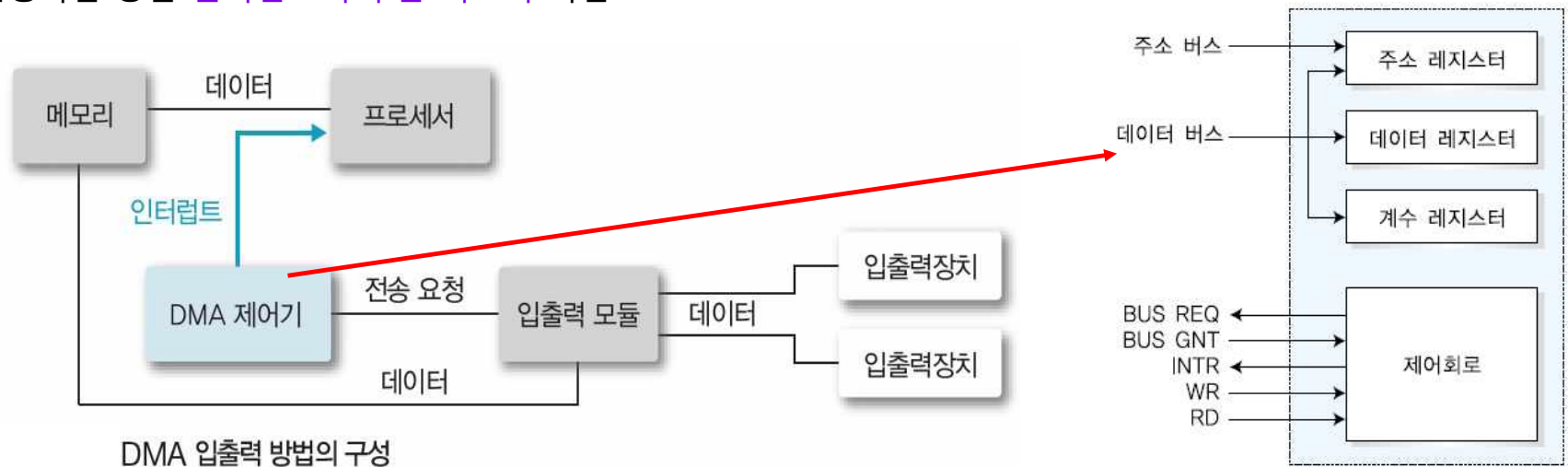
=>입출력 모듈이 인터럽트를 요구했을 때, 인터럽트 처리 과정

- 중앙처리장치는 인터럽트에 응답하기 전에 현재 실행 중인 명령어의 실행을 완료한다.
 - 중앙처리장치는 인터럽트를 검사하고 인터럽트 요구가 있다면 요구를 발생한 장치에 확인 신호를 보낸다.
 - 확인 신호를 받은 장치는 인터럽트를 요구한 요구 신호를 제거한다.
 - 중앙처리장치는 새롭게 시작될 프로그램으로 제어를 넘겨줄 준비를 한다. 먼저 프로그램 상태 단어(PSW)와 프로그램 카운터(PC) 내용을 스택(stack)에 저장한다.
 - 새로운 프로그램의 시작 주소를 프로그램 카운터에 적재한다. (인터럽트 서비스를 시작한다)
-
- 5단계 : 중앙처리장치는 입력된 데이터를 요구한다.
 - 6단계 : 입출력 모듈이 중앙처리장치로 데이터를 전송한다.
-
- 프로세서가 폴링 절차를 생략할 수 있어 입출력 연산의 대기시간이 필요 없으므로 효율성 증가
 - 현재의 작업 내용을 스택에 저장해야 하는 오버헤드가 발생하고, 프로세서의 레지스터에서 모든 데이터 전송한다는 단점

입출력 제어 방법

■ DMA Direct Memory Access 입출력 방법

- 프로세서의 도움 없이도 메인 메모리를 직접 제어하여 데이터를 전송하는 형태
- 직접 메모리 액세스라고 함
- 직접 기억장치 액세스(DMA, Direct Memory Access) 방식은 **대용량의 데이터를 이동시킬 때 효과적인 기술로** 기억장치와 입출력 모듈 간의 데이터 전송을 별도의 하드웨어인 DMA 제어기가 처리하고, 중앙처리장치는 개입하지 않도록 하는 방식
- 프로그램이 실행하는 동안 **인터럽트의 수를 최소화** 시킴
- 기본 구성

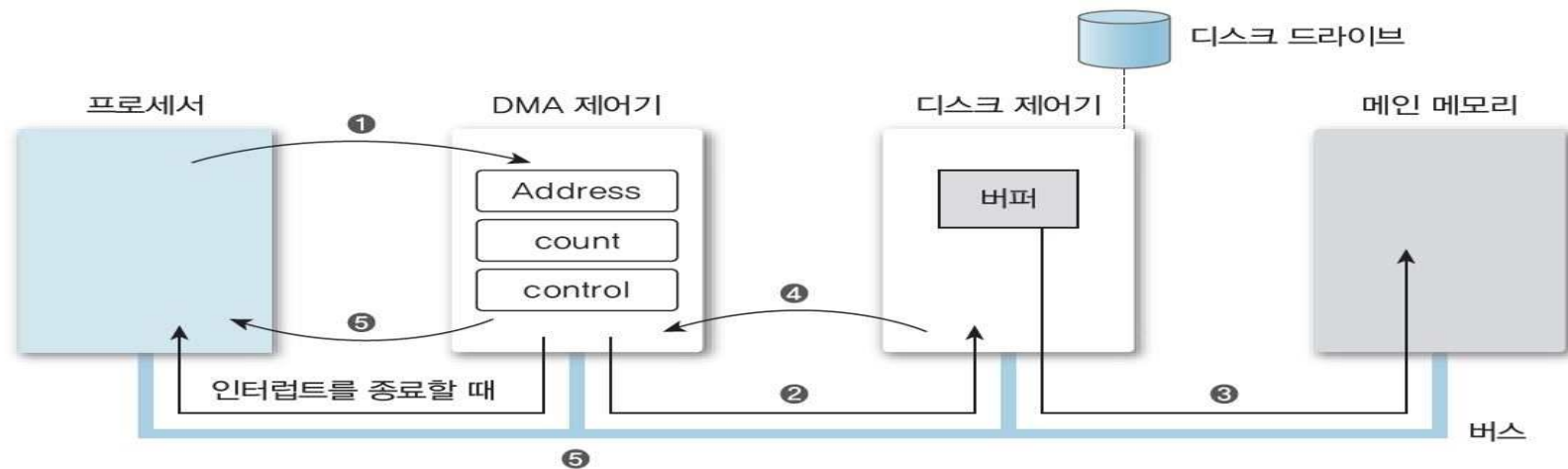


- 프로세서가 읽기와 쓰기 정보, 입출력 주소와 메모리 주소, 길이를 DMA 제어기에 전달하여 입출력을 요청하면 DMA 제어기는 이것을 직접 처리
- 입출력장치에서 메모리로 데이터를 블록 단위로 전송 가능

입출력 제어 방법

- 디스크의 데이터 전송과 멀티미디어 같은 대용량 데이터 전송에 적합
- DMA 전송 동작

프로세서는 데이터 전송을 시작하고 종료 할 때만 관여



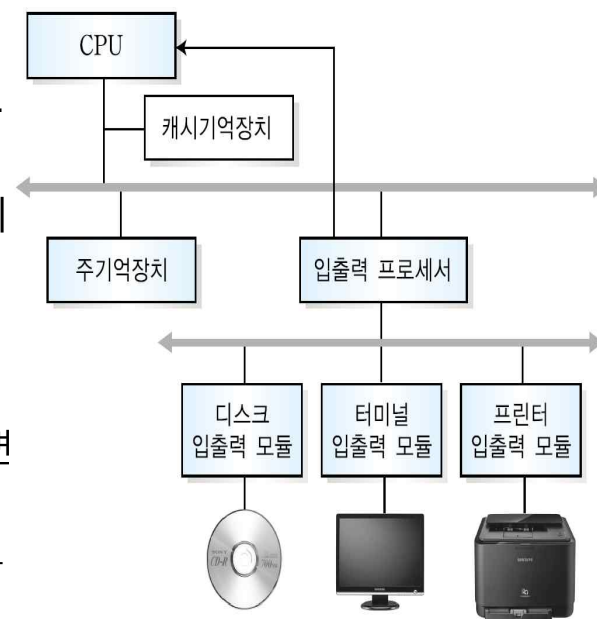
- ① 프로세서가 전송 방향, 전송 바이트 수, 데이터 블록의 메모리 주소 등을 DMA 제어기에 보낸다.
- ② DMA 제어기는 디스크 제어기에 데이터를 메인 메모리로 전송하라고 요청한다.
- ③ 디스크 제어기가 메인 메모리에 데이터를 전송한다.
- ④ 데이터 전송을 완료하면 디스크 제어기는 DMA 제어기에 완료 메시지를 전달한다.
- ⑤ DMA 제어기가 프로세서에 인터럽트 신호를 보낸다.

DMA 전송 동작

입출력 제어 방법

■ 입출력 채널(프로세서)를 이용한 입출력 제어방식

- 입출력 처리를 전담하는 별도의 입출력 프로세서(I/O Processor)를 두어, CPU의 효율을 높이는 입출력 제어방식이다.
- DMA의 개념을 확장한 입출력 전담 프로세서를 입출력 채널(I/O channel)이라고도 한다.
- 사용되는 입출력 프로세서(IOP)는 DMA 제어기의 기능을 향상시킨 것으로 입출력 명령 어들을 실행할 수 있는 프로세서이며, 데이터 블록을 임시 저장할 수 있는 지역 기억장치(local memory)를 포함하고 있다.
- 시스템 버스에 대한 인터페이스 및 버스 마스터 회로와 입출력 버스 중재 회로를 포함하고 있다.
- 중앙처리장치는 계산 업무에 필요한 데이터만을 처리하고, 입출력 프로세서는 여러 주변장치와 주기억장치 사이의 데이터 전송을 위한 통로를 제공한다.
 - 처음에는 중앙처리장치가 입출력 프로세서의 입출력 전송을 시작하도록 하지만, 그 이후에는 중앙처리장치와는 독립적으로 입출력 프로세서가 동작한다.
- 입출력 방식은 정도의 차이는 있으나 중앙처리장치의 간섭을 받게 되지만, 입출력 전용 프로세서를 이용해 간섭을 최소화하여 중앙처리장치의 이용 효율은 증가한다. 그러나 별도의 입출력 프로세서로 인한 하드웨어 비용이 증가하는 단점이 있다.



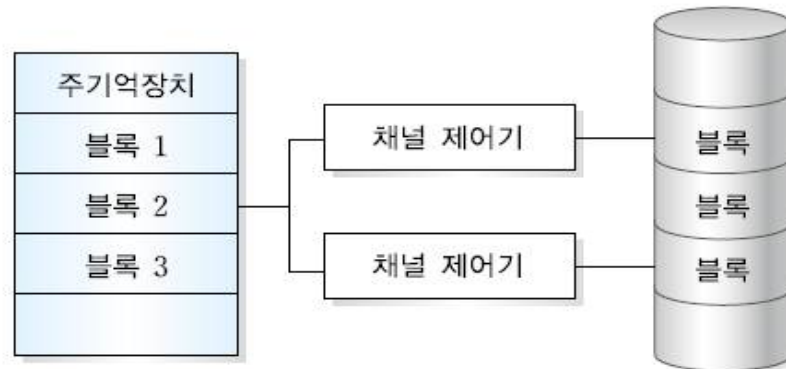
- 입출력 프로세서는 주변장치의 입출력 모듈을 제어해서 독립적으로 데이터를 송신할 수 있다.

입출력 제어 방법

■ 입출력 채널 프로그램

- DMA의 개념을 확장한 입출력 전담 프로세서를 **입출력 채널(I/O channel)**이라고도 한다.
- 입출력 채널이 여러 개의 블록을 입출력 할 때, 각 블록에 대한 채널 명령어의 모임을 **채널 프로그램**이라고 한다.
- 입출력 채널과 DMA는 모두 주기억장치에 접근하여 자율적인 입출력이 되도록 하므로 역할만 보면 거의 같다. 하지만 DMA는 하나의 블록을 입출력할 수 있으나, 입출력 채널은 블록을 여러 개 입출력 할 수 있다.

■ 입출력 채널을 이용하여 블록이 입출력 되는 개념

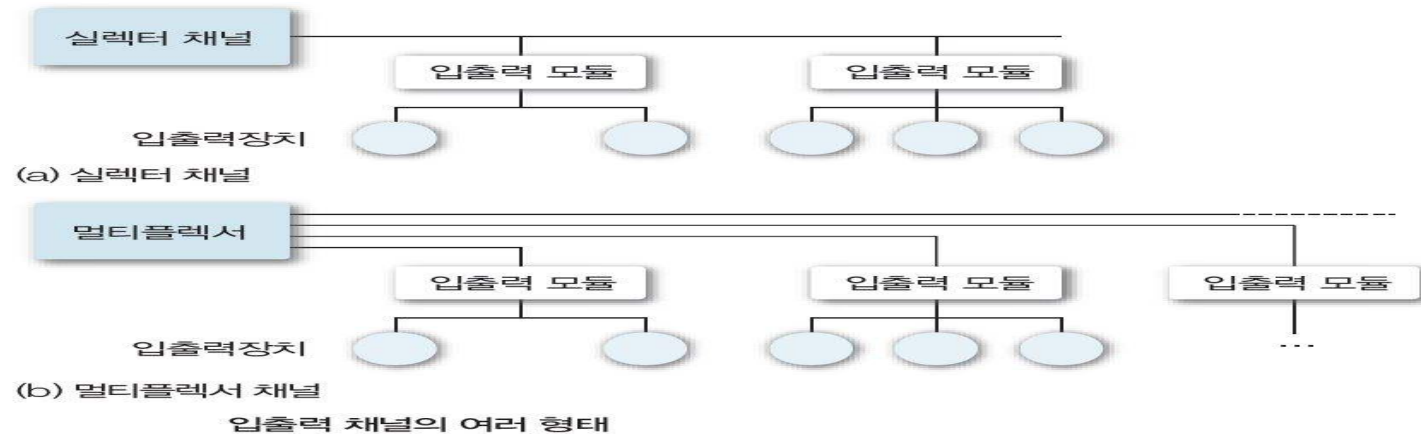


- 입출력 채널은 여러 개의 블록을 입출력할 수 있으므로 입출력 되는 모든 정보를 알려주어야 하지만, 입출력 명령 하나로 여러 블록에 대한 정보를 알려주는 것은 어렵다.
- 이를 해결하기 위해, 입출력 채널 제어기 내에 입출력 되는 모든 정보를 기억할 수 있도록 장비를 갖추 수 있으나 하드웨어 비용의 증가가 필수적

입출력 제어 방법

■ 채널의 종류(입출력 장치의 성질에 따라서)

- 셀렉터 채널(selector channel) : 어떤 장치의 입출력을 종료할 때까지 다른 장치를 실행하지 않도록 하는 것. 자기디스크, 자기테이프, 드럼과 같은 고속장치들은 셀렉터 채널을 사용하여 데이터를 고속으로 전송하기 때문에 다른 장치들과 다중화 곤란. 어느 한순간에 입출력 장치 중 한 장치와 하는 데이터 전송만 제어하므로 입출력 채널은 장치를 한 개 선택하여 데이터 전송
- 멀티플렉서 채널(multiplexer channel) : 바이트 단위로 시분할하여 여러 장치의 출력 처리. 동시에 여러 장치가 입출력 가능한 방식. 스위치 장치를 통해서 각 서브채널들이 순차적으로 연결되는 것이 멀티플렉서 채널의 원리이다. 다수의 저속·중속장치(카드 리더, 프린터)들을 연결할 때 사용. 채널과 메모리 연결은 장치와 채널 사이의 데이터 전달 속도보다 더 빠른 속도로 데이터를 전달할 수 있기 때문에 속도가 느린 다수의 입출력장치를 동시에 동작
- 블록 멀티플렉서 채널(Block multiplexer channel) : 셀렉터 채널과 멀티플렉서 채널의 장점을 결합한 방식으로 융통성 있는 운용을 할 수 있는 방식

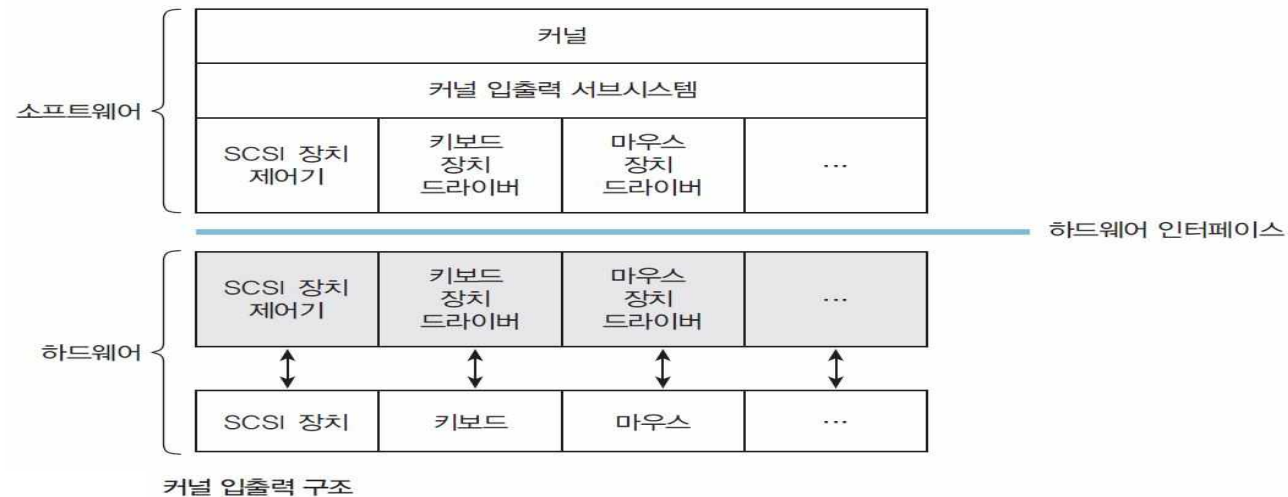


입출력 제어 방법

■ 커널 입출력 서브 시스템

■ 커널 입출력 구조

- 커널(운영체제)이 제공하는 입출력 서비스와 입출력 인터페이스를 입출력장치들이 정의한 방법에 따라 실행하려면 먼저 인터페이스를 구성해야 함
- 예를 들면 응용 프로그램은 입출력 장치의 모델이나 제조사와 관계없이 공통된 특징을 지닌 입출력 장치들을 액세스하는데 필요한 표준 함수를 정의한다. 그리고 정의된 함수는 키보드, 마우스 등 장치 드라이버로 수행한다. 장치 드라이버는 여러 입출력 장치의 차이를 간단한 표준 인터페이스로 변환하여 상위의 커널 입출력 서브 시스템에 제공한다.
- 커널 입출력 구조



입출력 제어 방법

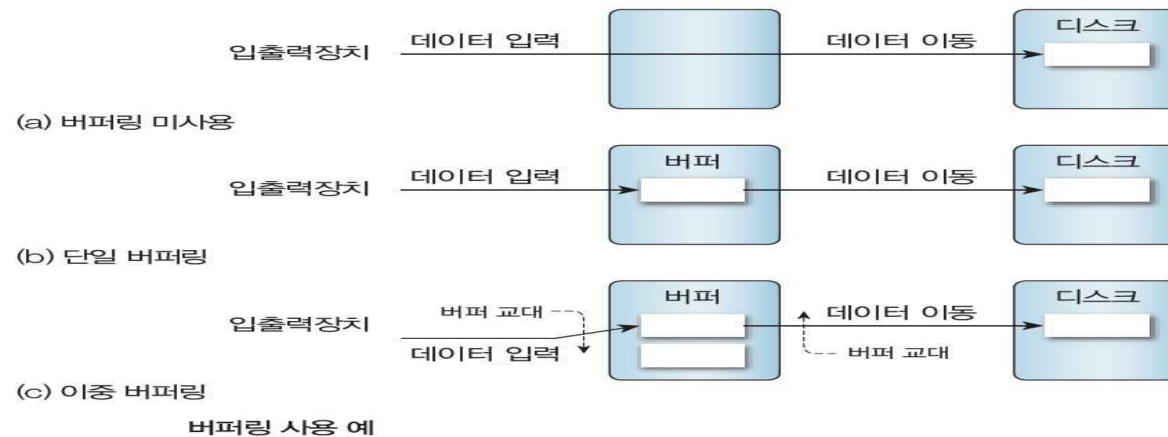
■ 커널이 제공하는 입출력 관련 서비스(컴퓨터의 효율성 증진)

■ 입출력 스케줄링

- 입출력 요구를 스케줄링한다는 것은 요구들의 실행 순서를 결정한다는 의미.
- 응용 프로그램은 입출력을 요청한 대로 실행 순서를 결정하기보다는 전반적인 시스템 성능을 항상 시키면서 프로세스의 요구를 공평하게 처리할 수 있도록 실행 순서를 결정한다. => 디스크 입출력 스케줄링 대표

■ 버퍼링

- 버퍼링은 입출력장치와 응용 프로그램 사이에 전송되는 데이터를 버퍼에 임시로 저장하는 방법
- 버퍼링은 송신자와 수신자의 전송속도 차이로 발생하는 시스템의 데이터 전송 문제를 해결



입출력 제어 방법

■ 커널이 제공하는 입출력 관련 서비스(컴퓨터의 효율성 증진)

■ 캐싱

- 캐싱은 명령어와 데이터를 캐시에 일시적으로 저장하여 프로세서와 메모리 간의 액세스 속도 차이를 줄여서 컴퓨터 성능 향상 방법
- 캐시는 버퍼와 달리 자주 사용할 자료를 미리 복사하여 저장하는 빠른 메모리 영역. 데이터가 위치하는 유일한 장소가 버퍼라고 한다면, 다른 곳에 저장된 데이터의 복사본을 저장하는 장소는 캐시
- 캐싱과 버퍼링은 서로 기능이 다르지만, 때로는 한 메모리 영역을 두 가지 목적에서 사용 가능.
- 데이터가 캐시에 있으면 디스크 입출력을 생략하여 입출력의 효율성 높임
- 메모리 내에 버퍼를 사용하여 디스크 입출력 스케줄링을 효율적으로 수행하거나 프로세스 간에 공유해야 하는 파일들을 위한 캐시로도 사용 가능

■ 스푼링

- 많은 응용 프로그램이 동시에 출력 데이터를 프린터로 전송하면 커널(운영체제)은 각 출력이 다른 프로그램의 출력과 섞이지 않고 프린터로 출력할 수 있도록 관리. 각 응용 프로그램의 출력은 각각 대응되는 디스크 파일에 스푼링한다. 스푼링 시스템은 응용 프로그램이 출력 데이터 작업을 종료하면 그때까지 모아 놓은 출력 데이터를 프린터 출력용 준비 큐에 삽입하여 스푼 작업을 한 번에 하나씩 프린터에 내보낸다. 이런 스푼링은 커널 스레드로 처리

입출력 제어 방법

■ 커널이 제공하는 입출력 관련 서비스(컴퓨터의 효율성 증진)

■ 오류 처리

- 커널은 입출력장치의 고장이나 네트워크 전송 오류로 발생하는 일시적인 고장을 효과적으로 해결, 디스크 읽기 실패나 망의 전송 오류는 재실행하여 문제 해결. 그러나 중요한 구성 요소가 영구적인 고장을 일으키면 커널이 문제를 완전히 극복하기는 곤란

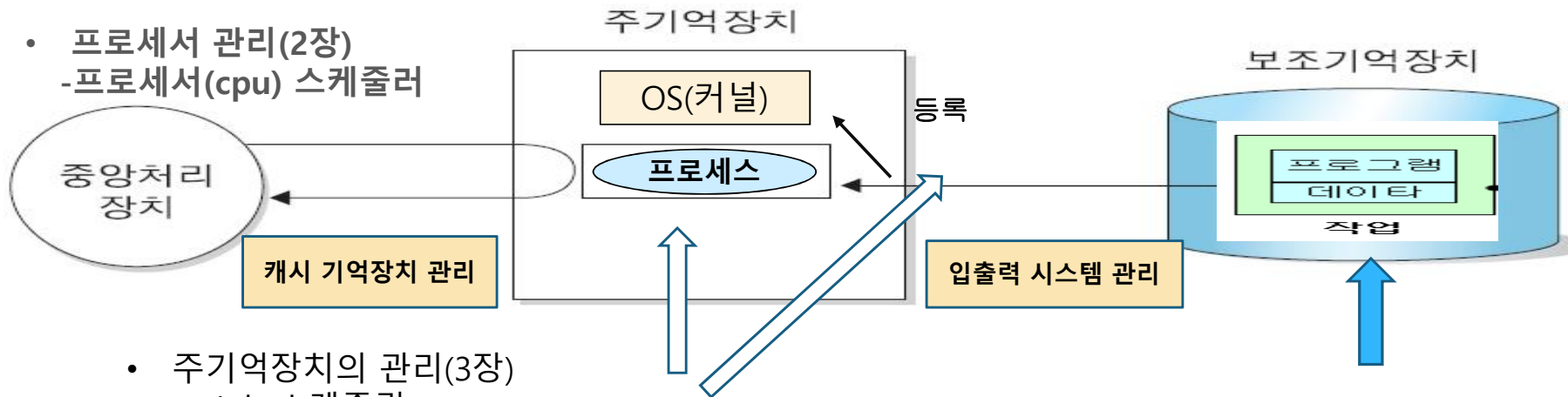
■ 자료 관리

- 커널은 입출력 구성에서 상태 정보를 유지. 입출력 서비스를 커널이 아닌 독립 프로그램이 담당하면 장치 드라이버와 커널의 정보 공유로 메시지를 교환하려는 오버헤드가 증가하는 단점. 장점은 입출력 시스템의 구조와 설계가 간단하고, 운영체제 커널의 크기가 작고, 유연성이 좋다는 것.

■ 입출력 서브시스템은 광범위한 서비스를 조정하며, 다음 사항들을 관리

- 장치 이름과 장치 액세스 제어 관리
- 장치 할당과 입출력 스케줄링 관리
- 버퍼링, 캐싱, 스푼링 관리
- 장치 상태 모니터링과 오류 처리, 고장 복구 관리

5장 디스크 스케줄링과 파일 시스템



- 주기억장치의 관리(3장)
 - Job 스케줄링
 - 주소 바인딩
 - 주기억장치 관리 기법
 - => 인출 기법, 배치 기법, 교체 기법, 할당 기법(연속)
- 가상 메모리 관리(4장)
 - 분산 할당 기법
 - => 페이징 기법, 세그먼테이션 기법
 - 페이지 교체 기법
 - => FIFO(First In First Out),
OPT(Optimal Replacement), LRU, LFU, NUR, SCR

- 디스크 스케줄링 (5장)
 - FCFS, SSTF, SCAN, C-SCAN
 - => 탐색시간 최소화
- 파일 시스템(5장)
 - 파일 구조
 - 디렉토리 구조