파일시스템과 입출력장치

연세대학교

파일시스템

연세대학교

디스크의 논리적 구조

연세대학교

- 논리적 블록
  - 운영체제에서 디스크는 1차원 블록의 배열로 다루어짐
    - 논리적 블록 번호: 0부터 N-1까지
  - 대개 블록의 크기는 섹터의 크기 (보통 512KB) 또는 섹터의 배수의 크기이다.
- 물리적 블록 맵핑
  - 운영체제는 논리적 블록들을 실린더, 트랙, 섹터 번호로 지정되는 물리적 블록에 맵핑한다.
  - 물리적으로 인접한 블록을 논리적으로도 인접하도록 맵핑한다.

파일 (File)

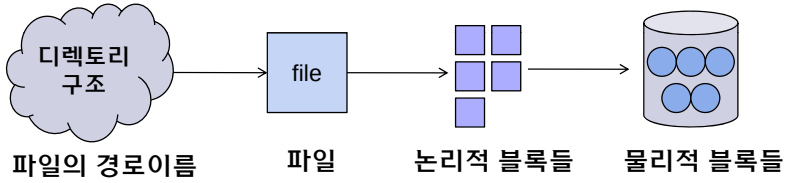
연세대학교

- 정보는 다양한 종류의 보조저장장치에 저장된다.
  - 디스크, 테이프, 플로피디스크, CD-ROM 등
- 운영체제는 다양한 저장 매체에 대한 균일한 논리적 뷰(view)를 제공한다. → 보조저장장치의 추상화
- 파일 (file)
  - 파일은 정보의 논리적 저장 단위로서 이름이 부여됨
  - 파일은 유형에 따라서 여러 가지 구조를 가지고 있다.
  - 파일을 사용함으로써 사용자에게 저장매체의 물리적 성질을 숨기고 저장매체를 추상화한다.
- 파일 시스템 설계
  - 파일 시스템이 사용자에게 보여지는 방법을 정의
  - 파일 시스템을 물리적 저장 매체에 맵핑하는 알고리즘 설계

파일 시스템의 계층적 설계

연세대학교

- 디렉토리 구조
  - 많은 파일들을 체계적으로 관리하기 위해서 디렉토리 구조를 사용함
- 파일의 물리적 저장 매체에 맵핑



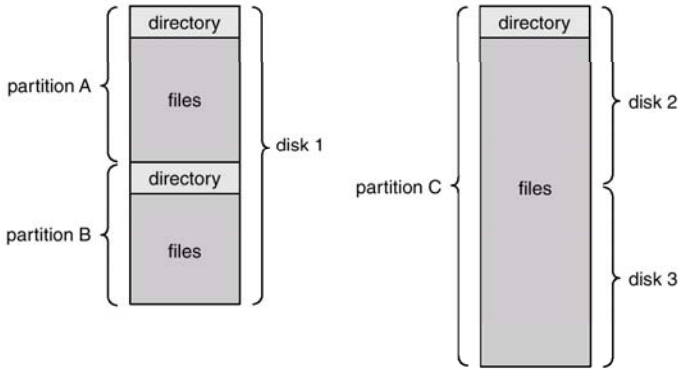
파일 시스템의 구조

연세대학교

- 파일 시스템 구조
  - 디스크는 여러 파티션(partition)으로 분할된다.
    - 여러 디스크를 하나의 파티션으로 사용하는 경우도 있다.
  - 각 파티션에 파일 시스템이 생성된다.
  - 파일 시스템은 저장된 파일과 저장된 파일에 대한 정보로 구성된다.
- 파일 시스템에 저장된 파일에 대한 정보는 디렉토리 구조를 사용하여 체계적으로 관리된다.

파일 시스템의 구조(2)

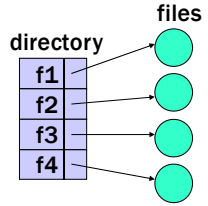
연세대학교



디렉토리

연세대학교

- 디렉토리(directory)
  - 파일에 대한 목록을 가진 자료 구조
  - 파일 이름을 디렉토리 엔트리로 변환하는 심볼테이블 역할
  - 디렉토리 엔트리는 저장된 파일에 대한 정보를 알려줌

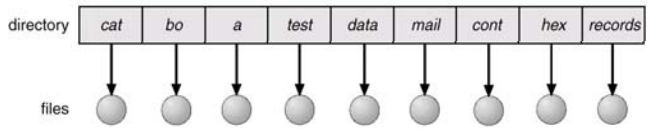


- 디렉토리 구조
  - 모든 파일에 대한 정보를 포함하고 있는 디렉토리들의 집합에 대한 구조

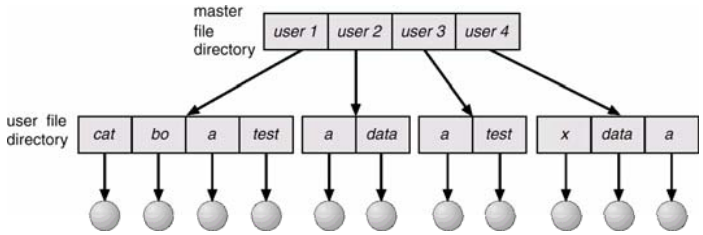
디렉토리의 논리적 구조

연세대학교

- 단일 단계 디렉토리 - naming, grouping 문제



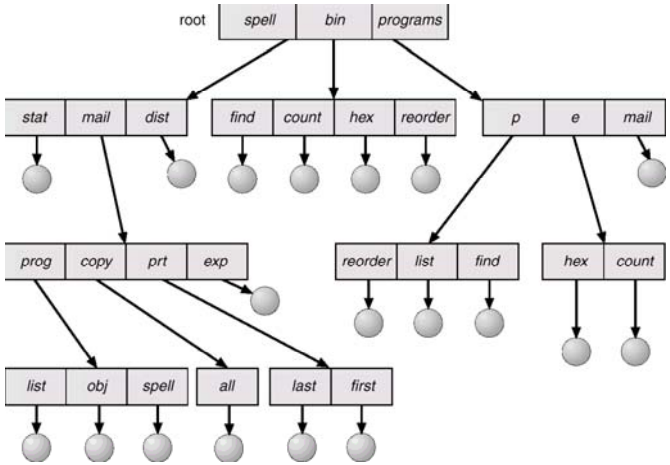
- 2단계 디렉토리 – 경로(path) 개념 등장



디렉토리의 논리적 구조(2)

연세대학교

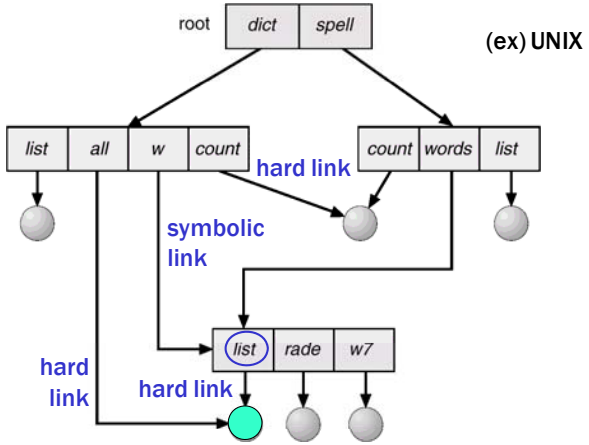
- 트리 구조 디렉토리 – MS-DOS



디렉토리의 논리적 구조(3)

연세대학교

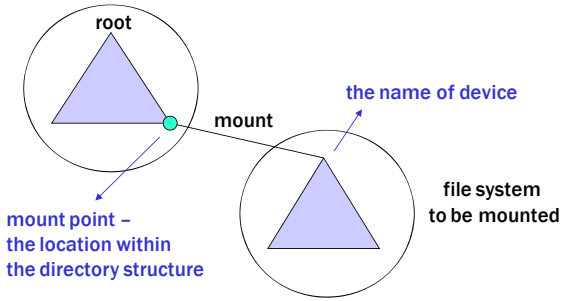
- 비순환 그래프 구조 디렉토리



파일 시스템 마운팅

연세대학교

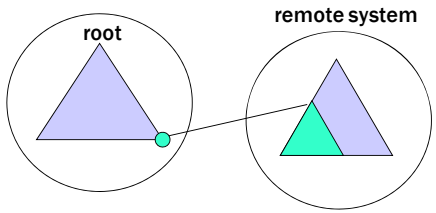
- 파일 시스템은 사용하기 전에 mount 되어야 한다.
- 운영체제는 마운트 할 장치의 이름과 디렉토리 구조 내의 마운트 할 위치를 받아서 장치를 지정된 위치에 마운트 한다.
  - (cf) 윈도우는 내 컴퓨터에 장치이름으로 마운트된다.
- 마운트를 통하여 여러 장치 또는 파티션들이 하나의 디렉토리 구조에 통합되어 사용될 수 있다.



파일 공유

연세대학교

- 다중 사용자 환경에서 여러 사용자가 파일을 공유하여 사용할 수 있다.
- 파일 공유는 보호 메커니즘을 거쳐서 이루어진다.
- 원격 파일 시스템
  - 다른 컴퓨터의 파일들을 네트워크를 통하여 공유할 수 있다.
  - 원격 컴퓨터의 파일 시스템의 일부분을 지역 파일시스템에 마운팅을 하여 사용할 수 있다.
    → 네트워크 파일 시스템 또는 분산 파일 시스템



파일 보호

연세대학교

- 파일에 대한 보호는 접근 유형을 제한하여 수행한다.
- 접근 유형
  - 읽기
  - 쓰기
  - 실행
  - 첨가(append): 파일 끝에 추가
  - 삭제
  - 파일 목록 열람
- UNIX의 접근 제어
  - 소유자(user), 그룹(group), 나머지 사용자(others)별로 접근 유형을 지정한다.
- Window 접근 제어
  - 사용자 별로 접근 유형을 지정한다.

7.3 I/O 장치의 접속

연세대학교

- I/O 제어
  - I/O 장치는 시스템 버스와 직접 접속되지 않고 인터페이스 장치인 I/O 제어기를 통하여 접속된다.

- I/O 장치가 시스템 버스에 직접 접속되지 못하는 이유
  - I/O 장치들의 데이터 전송 속도가 CPU의 데이터 처리 속도에 비하여 훨씬 더 느리다. 따라서 고속의 시스템 버스와 I/O 장치들 사이에 직접 데이터를 교환하는 것은 불가능
  - I/O 장치들은 종류에 따라 제어 방법이 서로 다르며 별도의 제어 회로가 필요함
  - I/O 장치들과 CPU가 사용하는 데이터 형식의 길이가 서로 다른 경우가 많음

인터럽트와 입출력

연세대학교

인터럽트

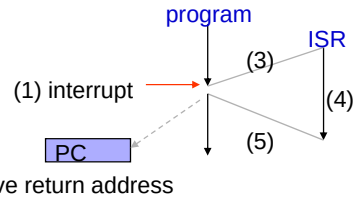
연세대학교

- 인터럽트(Interrupt)
  - 프로그램 실행 중에 CPU의 현재 처리 순서를 중단시키고 요청한 다른 동작을 수행하도록 하는 시스템 동작
- 인터럽트의 종류
  - 하드웨어 인터럽트
    - 외부 입출력 장치에 의해서 하드웨어적으로 인터럽트 요청
  - 내부 인터럽트
    - CPU의 명령어 실행 동안 오류가 발생했을 때에 이에 대한 처리를 위해 CPU 내부적으로 인터럽트 요청(0으로 나누기, 주소 오류, 보호 오류 등)
    - "예외", "트랩" 이라고도 부름
  - 소프트웨어 인터럽트
    - 운영체제의 서비스를 받기 위해서 기계어 명령어를 사용하여 인터럽트 요청
    - "시스템 호출" 이라고도 부름

인터럽트 처리 과정

연세대학교

- 인터럽트 처리 과정
  - 인터럽트 요청을 받으면 (1)
    - 현재 수행 중인 명령어 수행을 끝내거나 멈추고,
    - 현재의 PC 값을 저장하고 (2)
    - 인터럽트 서비스 루틴(Interrupt Service Routine)으로 분기 (3)
  - 인터럽트 서비스 루틴의 수행을 끝마치면 (4)
    - PC를 저장된 PC값으로 복원하여 인터럽트 요청 당시의 주소로 분기하여 중단된 프로그램 실행을 재개함 (5)



운영체제와 인터럽트

연세대학교

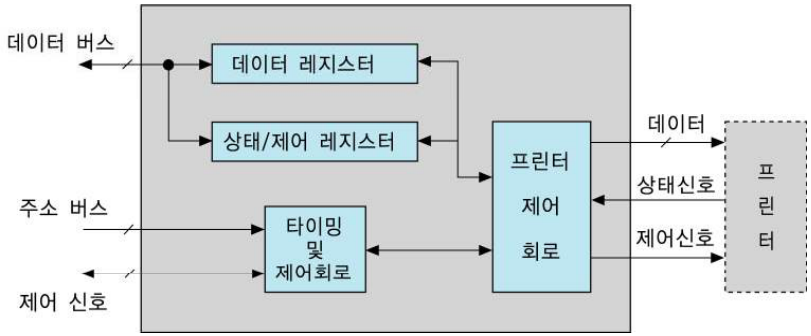
- 운영체제 프로그램은 인터럽트가 발생할 때에만 수행한다.
  Operating System is Interrupt Driven
- 운영체제 작업
  - 입출력 서비스, 시분할 처리
    - 하드웨어 인터럽트에 의해서 수행
  - 오류 처리 :
    - 내부 인터럽트에 의해서 수행
  - 응용 프로그램에게 운영체제 서비스 제공
    - 소프트웨어 인터럽트에 의해서 수행

보호와 CPU 동작 모드

연세대학교

- 보호와 CPU 동작 모드
  - 운영체제를 사용자 프로세스로부터 보호하기 위해서 CPU는 두 가지 이상의 동작 모드를 갖고 있어야 함
- CPU의 이중 동작 모드
  - 커널 모드(시스템 모드, 특권 모드, 모니터 모드)
    - 운영체제가 수행 중일 때의 동작 모드
    - 특권 명령어를 사용할 수 있음
      - 시스템 제어, 입출력 작업 수행
  - 사용자 모드 :
    - 사용자 프로세스가 수행 중일 때의 동작 모드
    - 특권 명령어를 사용할 수 없음
      - 특권 명령어를 실행하면 보호 오류로 내부 인터럽트 발생
- 인터럽트가 발생하면 CPU 동작 모드는 커널 모드로 전환됨
  - 인터럽트 서비스 루틴은 운영체제 프로그램임

7.3 프린터 제어기의 내부 구성도



상태 레지스터와 제어 레지스터

- 상태 레지스터 (대개 읽기전용)
  - I/O 장치의 상태와 오류 검사 결과 등을 나타내는 비트들로 구성
- 제어 레지스터 (대개 쓰기전용)
  - CPU가 보낸 I/O 명령(command)을 저장
- 두 레지스터는 대개 하나의 주소를 사용함

- CPU가 프린터로 데이터를 출력하는 과정
  1. CPU가 프린터 제어기에게 프린터의 상태를 검사하도록 요청
  2. 제어기는 프린터의 상태를 검사하여 CPU에게 통보:
  3. 데이터를 받을 준비가 된 상태면, CPU는 제어기에게 출력 명령과 데이터를 전송
  4. 제어기는 프린트 동작을 위한 제어 신호와 함께 데이터를 프린터로 전송

I/O 수행 방법

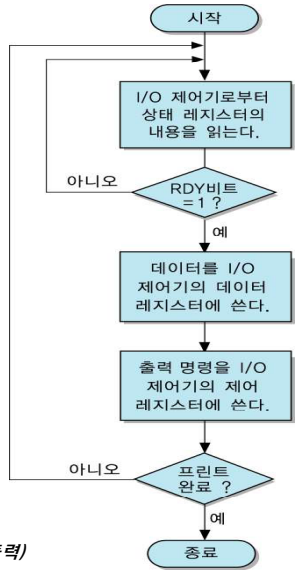
- 프로그램 이용 I/O (polling 방식)
- 인터럽트 이용 I/O
- DMA

프로그램 이용 I/O (programmed I/O)

- CPU가 반복적으로 I/O 장치의 상태를 검사하면서 I/O 동작을 처리하는 방식

- 장점
  - 간단하며, 별도의 하드웨어가 필요하지 않음
- 단점
  - CPU가 I/O 동작에 직접 관여해야 하므로, 그 동안에 다른 일을 하지 못함

프로그램을 이용한 I/O의 흐름도



I/O 주소지정(I/O addressing)

- 각 I/O 장치 당 두 개(이상)씩의 주소 할당
  - 데이터 레지스터 주소
  - 상태/제어 레지스터 주소
- I/O 주소 지정 방법
  - 기억장치-사상 I/O(memory-mapped I/O)
    - 메모리 주소의 일부를 입출력 장치에 제공
    - 메모리 읽기/쓰기 명령어(LOAD, STORE)를 입출력에도 사용
  - 분리형 I/O(isolated-I/O)
    - 메모리 주소와 입출력 주소 공간이 분리되어 있음
    - 입출력 장치는 입출력 주소를 사용
    - 입출력용 명령어(IN, OUT)를 별도로 제공함

7.4 인터럽트를 이용한 I/O

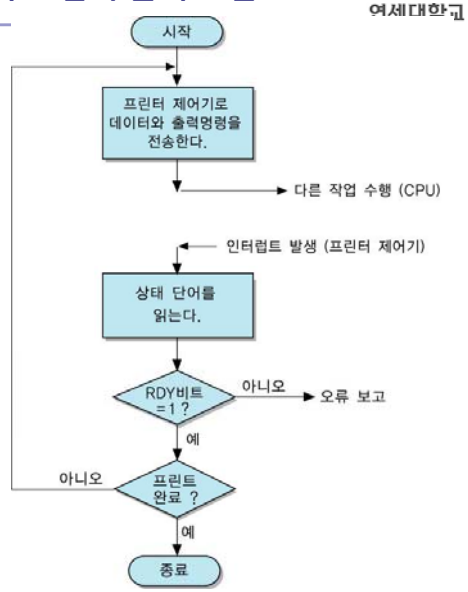
- 인터럽트-구동 I/O(interrupt-driven I/O) :
  - 인터럽트 메커니즘을 이용함으로써, I/O 동작이 진행되는 동안에 CPU가 다른 작업을 처리할 수 있도록 하는 방식
- 동작 순서
  - CPU가 I/O 제어기에게 명령을 전송하고, CPU는 다른 작업을 수행
  - 제어기는 I/O 장치를 제어하여 I/O 명령을 수행
  - I/O 명령 수행이 완료되면, 제어기는 CPU로 인터럽트 신호를 전송
  - CPU는 인터럽트 신호를 받는 즉시 원래의 프로그램으로 복귀하여 수행을 계속

인터럽트-구동 I/O 방식에서의 프린터 출력 흐름도

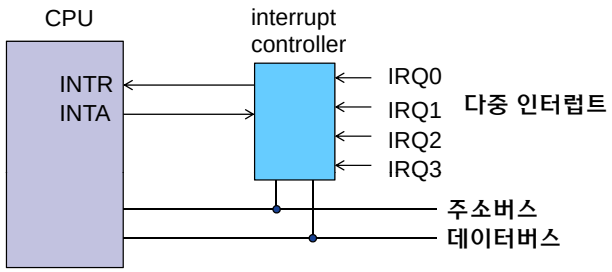
① CPU는 데이터와 프린트 명령을 프린터 제어기로 전송하고, 다른 작업을 수행

② 그 데이터의 프린트가 종료되면, 제어기가 CPU로 인터럽트 요구 신호를 전송

③ 프린트할 내용이 남아 있다면, CPU는 다음에 프린트할 데이터를 준비하여 위의 과정을 반복



인터럽트 회로 구성 예

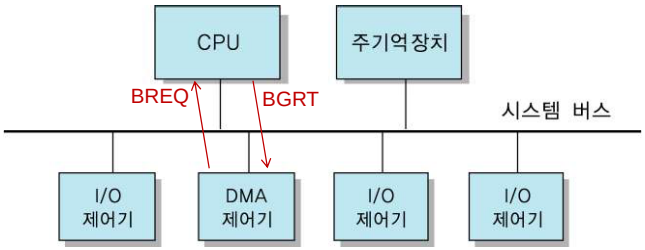


인터럽트요구(interrupt request:INTR)
인터럽트응답(interrupt acknowledge:INTA)

7.5 직접기억장치액세스

- Direct Memory Access (DMA) :
  - CPU의 개입 없이 I/O 장치와 기억장치 사이에 데이터를 전송하는 방식 (사이클 스틸링(cycle stealing)이라고도 함)
- 방법 :
  - CPU가 주기억장치를 액세스하지 않는 시간(CPU가 내부적으로 명령어를 해독하거나 ALU 연산을 수행하는 시간) 동안에 시스템 버스를 사용하여 주기억장치와 I/O 장치(디스크 제어기) 간에 데이터 전송

DMA 제어기가 포함된 시스템 구성도

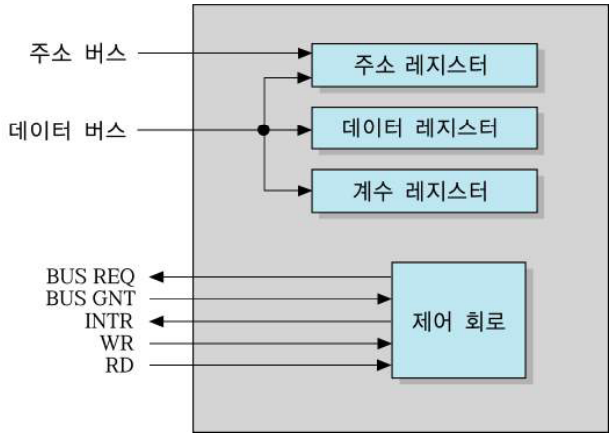


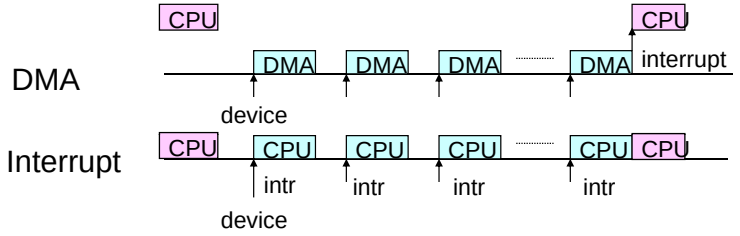
DMA 처리 순서

1. CPU가 DMA 제어기(DMA controller)로 아래 정보를 포함한 명령을 전송
   - I/O 장치의 주소
   - 연산(쓰기 혹은 읽기) 지정자
   - 데이터가 읽혀지거나 쓰여질 주기억장치 영역의 시작 주소
   - 전송될 데이터 단어들의 수
2. DMA 제어기는 CPU로 버스 요구(BUS REQ) 신호를 전송
3. CPU가 DMA 제어기로 버스 승인(BUS GRANT) 신호를 전송
   - 이 신호를 받으면 DMA는 새로운 bus master가 됨
4. DMA 제어기가 주기억장치로부터 데이터를 읽어서 디스크 제어기에 저장
5. 전송할 데이터가 남아있으면, 2번부터 4번까지를 다시 반복
6. 모든 데이터들의 전송이 완료되면 CPU로 INTR 신호를 전송

DMA 제어기의 내부 구조

연세대학교



DMA와 Interrupt 동작 비교

연세대학교