Slide 0

4부 입출력 시스템
(13장. 입출력 시스템)

개요

Slide 1

- 컴퓨터의 주요 두가지 작업
  - 입출력
  - 처리 (processing)
- 입출력 서브시스템
  - 등장배경
    - 기능과 속도측면에서 다양한 장치가 존재
    - 입출력 장치관리가 매우 복잡함
    - 커널에서 분리해서 입출력 서브시스템을 형성
  - 입출력 장치 기술 경향
    - 소프트웨어나 하드웨어 인터페이스의 표준화 증가
    - 다양한 입출력 장치의 등장
  - 장치 구동기(device driver)
    - 다양하고 상이한 장치에 대한 일관된 접근 인터페이스 제공
    - 운영체제는 장치 구동기를 통해 장치를 제어함

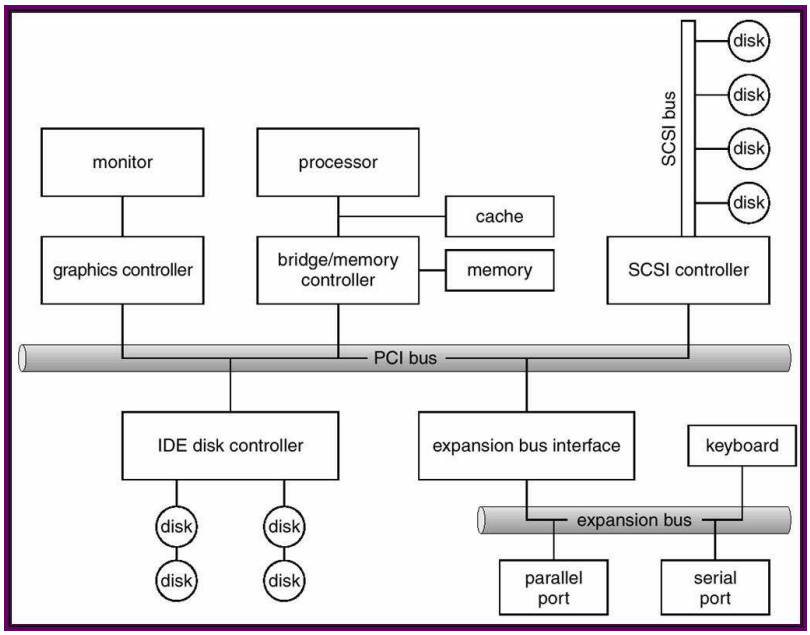
입출력 하드웨어

Slide 2

- 하드웨어
  - 장치
    - 저장장치: 디스크, 테이프등
    - 전송장치: 네트워크 카드, 모뎀등
    - 사용자 인터페이스 장치: 스크린, 키보드, 마우스등
  - 버스
    - 하나 이상의 장치들이 정보를 주고 받는 와이어 집합
    - 전형적인 PC 버스 구조

입출력 하드웨어 (계속)

Slide 3



입출력 하드웨어 (계속)

Slide 4

- 장치 제어
  - 프로세서 →장치
    - 장치제어기 내부의 레지스터에 명령을 기록
    - 필요한 기타 자료는 메모리 특정 위치 이용 가능
  - 장치 →프로세서
    - 명령 수행후 결과를 레지스터에 기록
    - 폴링 혹은 인터럽트로 프로세서에게 알림
  - 입출력 포트
    - status 레지스터: 제어기의 상태를 나타냄
    - control 레지스터: 제어기가 수행할 명령을 설정
    - data-in 레지스터: 장치에서의 데이터 입력용
    - data-out 레지스터: 장치로 데이터 출력용
  - 장치의 선택
    - 주소를 이용하여 여러장치 중 하나를 선택
    - I/O-mapped I/O 에서는 I/O 전용 명령어로 I/O 주소 공간 접속
    - memory-mapped I/O 에서는 메모리 명령어로 메모리 주소 공간 접속

입출력 하드웨어 (계속)

Slide 5

I/O address range (hexadecimal)	device
000-00F	DMA controller
020-021	interrupt controller
040-043	timer
200-20F	game controller
2F8-2FF	serial port (secondary)
320-32F	hard-disk controller
378-37F	parallel port
3D0-3DF	graphics controller
3F0-3F7	diskette-drive controller
3F8-3FF	serial port (primary)

입출력 하드웨어 (계속)

Slide 6

- 폴링(Polling)
  - 명령의 완료 및 데이터 수신을 장치제어기 상태레지스터를 읽어서 수행
  - 예)
    - 프로세서
      - 장치제어기 status 레지스터의 busy 비트가 해제될 때까지 기다림
      - busy 비트가 해제되면 command 레지스터에 write 비트 설정 후 data-out 레지스터에 데이터 기록
      - command 레지스터의 command-ready 비트를 설정
    - 장치 제어기
      - command-ready 비트 설정을 확인후 busy비트 설정
      - command 레지스터 해석 후 data-out 레지스터의 데이터를 장치로 출력
      - command-ready 비트를 해제후 status 레지스터에 명령 성공을 설정하고 busy 비트 해제
  - 문제점
    - 속도가 느린 장치에는 비효율적
    - 버퍼 오버플로우로 데이터 손실 발생 가능

입출력 하드웨어 (계속)

Slide 7

- 인터럽트
  - 설명
    - 명령의 완료및 데이터 수신을 인터럽트를 통해 프로세서에게 알림
    - 보통 인터럽트시 인터럽트 처리 루틴을 결정하기 위해 벡터번호를 사용함
    - 인터럽트 벡터 번호는 인터럽트 제어기가 제공 (부팅시 각 장치별로 설정됨)
  - 펜티엄 프로세서에서 인터럽트 벡터 테이블
  - 인터럽트 처리 과정

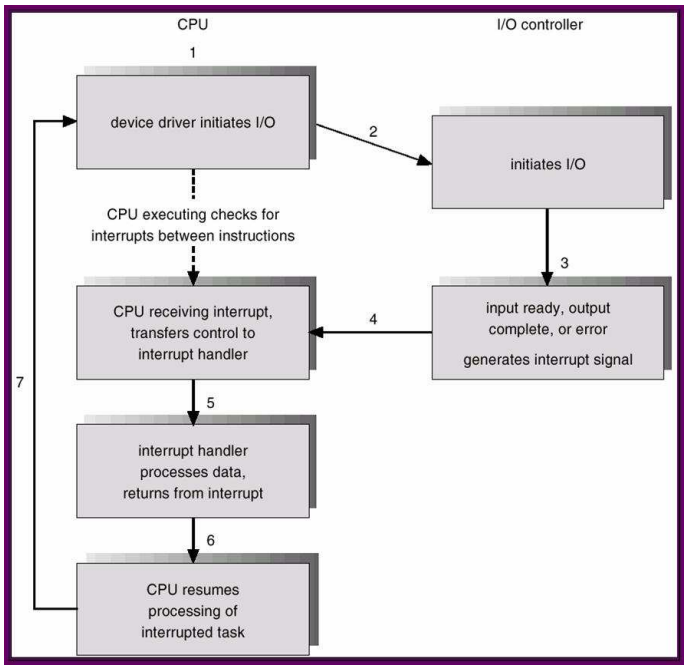
입출력 하드웨어 (계속)

Slide 8

vector number	description
0	divide error
1	debug exception
2	null interrupt
3	breakpoint
4	INTO-detected overflow
5	bound range exception
6	invalid opcode
7	device not available
8	double fault
9	coprocessor segment overrun (reserved)
10	invalid task state segment
11	segment not present
12	stack fault
13	general protection
14	page fault
15	(Intel reserved, do not use)
16	floating-point error
17	alignment check
18	machine check
19Ð31	(Intel reserved, do not use)
32Ð255	maskable interrupts

입출력 하드웨어 (계속)

Slide 9



입출력 하드웨어 (계속)

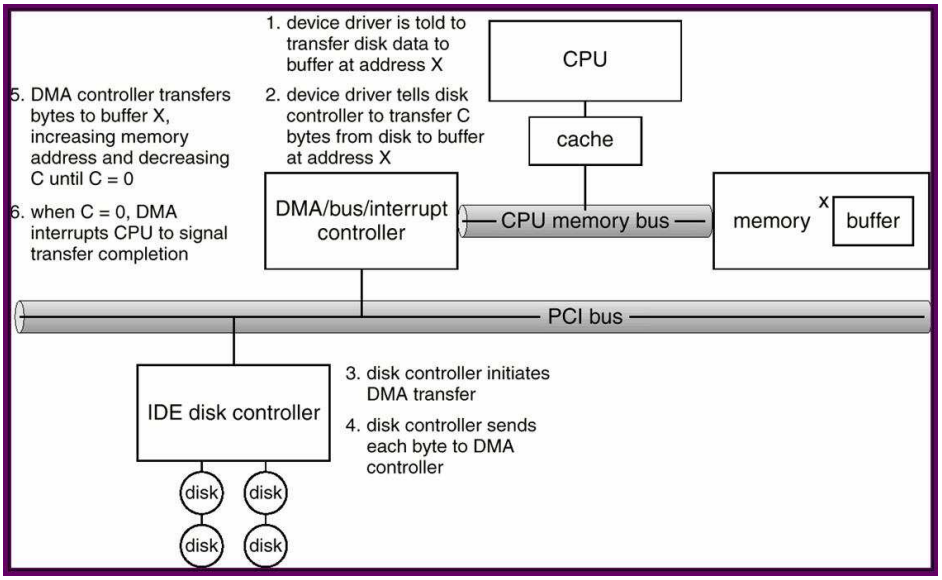
- 직접 메모리 접근 (DMA:Direct Memory Access)
  - 프로그램 입출력
    - 프로그램에 의하여 프로세서가 장치제어기와 데이터를 입출력하는 방법
    - 비싼 범용 프로세서가 하기에는 단순한 작업으로 낭비
    - 특히, 대량의 데이터 전송에는 비효율적임
  - DMA란 주기억장치와 IO 장치사이에 데이터를 전달해 주는 장치

Slide 10

  - 처리절차
    - 프로세서
      - 프로세서는 메모리에 DMA 수행 명령어 및 관련정보를 기록
        (관련정보에는 소스 포인터, 목적지 포인터, 전송 바이트 크기등이 있음)
      - 프로세서는 DMA 수행 명령어 주소및 명령을 DMA 제어기 내부에 기록
      - 프로세서 자신의 작업을 수행
    - DMA 제어기
      - 버스를 요청하여 획득하고 DMA 수행 명령을 해독하여 수행
      - 전송이 종료되면 인터럽트로 프로세서에게 알려줌
      - 장치 제어기와의 핸드쉐이킹은 DMA-request, DMA-acknowledge로 수행

입출력 하드웨어 (계속)

Slide 11



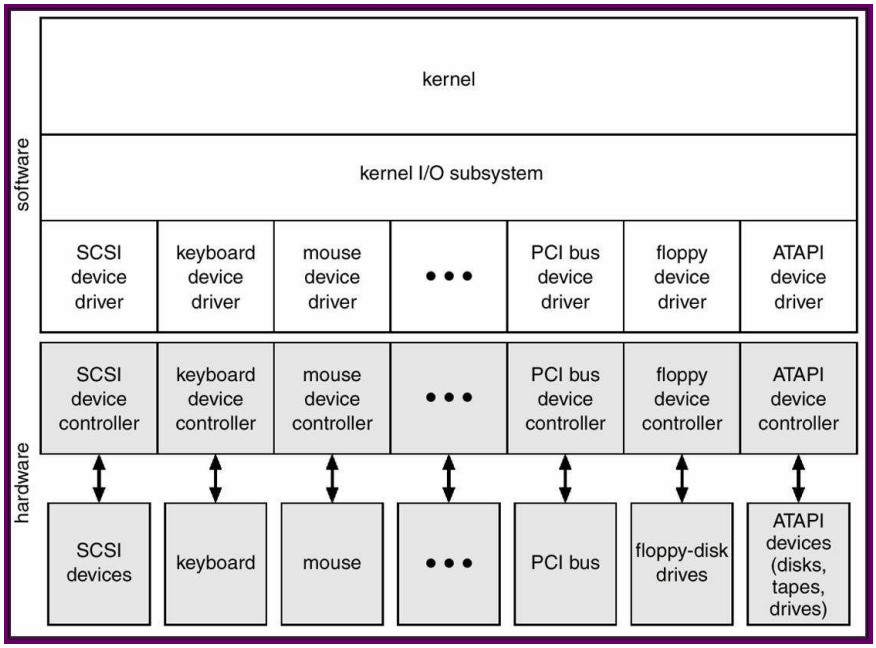
응용프로그램 입출력 인터페이스

Slide 12

- 운영체제와 장치간의 입출력
  - 문제: 다양한 특성을 가진 다양한 장치들을 어떻게 운영체제가 관리하나?
  - 해결
    - 추상화, 캡슐화, 계층화의 접근방법 사용
    - 규격화된 함수들의 집합을 정한 표준 인터페이스를 통하여 접근
    - 운영체제와 장치사이의 인터페이스 →장치 구동기 사용
  - 장치구동기(device driver)의 목적
    - 커널의 입출력 부시스템이
    - 각 장치의 특성에 상관없이
    - 일관된 방법으로 장치를 제어할 수 있도록 함
  - 장치구동기의 장점
    - 운영체제 개발자: 하드웨어를 고려하지 않고 입출력 부시스템을 설계
    - 하드웨어 업체: 운영체제에대한 세부적인 고려없이 하드웨어 설계 가능
    - 단, 입출력 부시스템과 장치관리자와의 인터페이스의 표준화 필요
    - 하드웨어 업체는 표준화된 방법에 따라서 장치구동기를 만들어 하드웨어와 함께 제공

응용프로그램 입출력 인터페이스 (계속)

Slide 13



입출력 기타이슈

Slide 14

- 두 종류의 장치
  - 문자 기반 장치 (character-oriented device)
    - 문자 단위로 입출력
    - 예) 키보드
  - 블록 기반 장치 (block-oriented device)
    - 블록 단위로 입출력
    - 예) 디스크
- 네트워크 장치: 소켓(socket) 인터페이스를 사용
- 클럭과 타이머
  - 현재시간 제공 (특수 칩으로 제공)
  - 경과시간 제공 (PIT 로 제공)
  - T시간에 X 동작을 발생시키는 시간 지정 (PIT 로 제공)

입출력 기타이슈 (계속)

Slide 15

- 블록킹과 논블록킹 입출력
  - 블록킹
    - 입출력 시스템 호출이 실제로 입출력이 종료된뒤 리턴됨
    - 응용프로그램은 대기 큐로 이동 됨
  - 논블록킹
    - 입출력 시스템 호출이 바로 리턴 됨
    - 응용프로그램은 중단되지 않음
- 스케줄링
  - 입출력 요구가 많아 대기 큐에 쌓였을때 스케줄링으로 성능향상 가능
  - 예) 디스크 스케줄링

입출력 기타이슈 (계속)

Slide 16

- 버퍼링
  - 사용 이유
    - 두 장치사이의 속도 불일치가 일어날때 사용
    - 두 장치사시의 데이터 전송 크기가 다를때 사용
- 캐싱
  - 느린 장치를 위해 사용
  - 느린 장치에서 읽은 일부의 데이터를 빠른 메모리에 유지
- 스풀링
  - 프린터나 테이프같은 장치로의 입출력시 데이터를 디스크 파일에
  - 일관되게 유지하여 동작시키는 방법
- 입출력 요청에 대한 하드웨어 작업

입출력 기타이슈 (계속)

Slide 17