

chapter.3 운영체제

컴퓨터 시스템의 구조

- 내부장치 - CPU,메모리
- 외부장치 - 디스크, 키보드, 마우스, 모니터, 네트워크 장치

외부장치에서 내부장치로 데이터를 읽어와 각종 연산을 수행한 후, 외부장치에서 내부장치로 데이터를 읽어와 연산을 수행한 후 다시 내보내는방식

- 입력(input) - 내부로 데이터가 들어오는 것
- 출력(output) - 외부로 데이터가 나가는것
- 입출력(I/O) - 컴퓨터 내부에서 연산을 한 후 디스크에 데이터를 저장한것

운영체제 - 여러 프로그램을 동시에 수행되는 시스템을 위한 것

전체를 메모리에 상주 하면 메모리 낭비가 발생

항상 메모리에 올라가 있는 부분은 전체 운영체제 중 핵심 부분에 한정 이러한 부분을 (커널 kernel) 이라 한다.

CPU 연산과 I/O 연산

- I/O 연산은 입출력 컨트롤러가 담당, 컴퓨터 내에서 수행되는 연산은 메인 CPU 가 담당한다.
- 입출력 장치와 메인 CPU 는 동시 수행 가능.
- 장치마다 들어오고 나가는 데이터를 로컬버퍼라는 작은 메모리에 저장.
- 인터럽트라인(interrupt line) - 컨트롤러들이 CPU의 서비스가 필요할때 이를 통보하는 방법

CPU는 명령 하나를 수행할 때마다 인터럽트가 발생되었는지 확인

인터럽트의 일반적 기능

인터럽트 처리루틴

- 각각의 인터럽트(interrupt vector)에 따라서 다르며 운영체제 개발자가 미리 구현하여 커널에 포함해둔다.

디스크 인터럽트

- CPU가 인터럽트의 코드를 수행 할당받은 경우 다음 명령을 수행 할 수 있음을 표시
- 하드웨어 인터럽트
 - 컨트롤러 등 하드웨어 장치가 CPU의 인터럽트 라인을 세팅
 - 인터럽트 종류 번호에 따른 처리 코드가 위치한 부분을 가리키는 자료구조
 - 실제 처리 코드는 인터럽트 처리 루틴(interrupt service routine) 또는 인터럽트 핸들러(interrupt handler)라고 불리는 곳에 정의

소프트웨어 인터럽트

- 트랩(trap) 이라는 용어로 주로 불림
- 예외상황 - 0으로 나누거나, 메모리 영역 바깥에 접근하려는 시도 등
- 시스템콜 - 운영체제에 서비스를 요청하는 경우로 화면출력, 키보드 입력 등의 커널 코드를 호출하여 CPU 제어권을 넘겨야하는 경우

인터럽트 랜들링

인터럽트가 발생한 경우에 처리해야 할 일의 절차를 의미

- CPU에서 명령이 실행될 때 CPU 내부에 있는 임시 기억 장치인 레지스터에 데이터를 읽거나 쓰면서 작업하며 미리 저장함
- 프로그램을 관리하는 PCB에 현재 프로그램의 실행 상태를 저장하고 CPU의 제어권이 넘어감
- 다시 실행될 경우 PCB에서 상태를 복원하고 실행
- 운영체제는 인터럽트가 발생할 때 실행

입출력구조

시스템이 컴퓨터 외부의 입출력 장치들과 데이터를 주고받는 것

동기식 입출력(Synchronous I/O)

- 프로그램이 입출력을 요청을 했을 때 입출력 작업이 완료된 후에야 프로그램이 후속 작업을 수행
- CPU가 입출력 연산이 끝날 때까지 인터럽트를 기다리며 자원을 낭비
- CPU 연산은 빠르지만 입출력 연산은 느리므로 CPU를 사용하지 못한다.
- 한번에 하나의 입출력만 수행

비동기식 입출력(asynchronous I/O)

- CPU 제어권을 입출력 연산을 호출한 프로그램에게 바로 다시 부여하는 것
- 입출력 데이터와 상관없이 다음 명령이 수행될 수 있는 경우는 입출력 연산과 상관없이 바로 수행하고 결과가 필요한 연산일 경우에만 기다렸다가 수행
- 쓰기 작업인 경우 다음 명령어 수행이 가능하므로 비동기식 입출력이 사용

DMA

- 메모리는 CPU에 의해서만 접근 할수 있다.
- 이럴경우 메모리 접근 연산이 모두 CPU의 인터럽트를 통해서 이루어지므로 효율성이 낮아짐
- 위 문제를 해결하기 위한 컨트롤러로 메모리에 접근할 수 있는 장치
- 로컬버퍼에서 데이터를 읽어오는 작업을 DMA가 CPU 대신 대행함
작업을 완료하면 CPU에 인터럽트로 알려주고 블록(block)단위로 읽어옴
- 인터럽트의 빈도를 줄여 효율적으로 관리 입출력연산을 빠르게 수행가능

저장장치 구조

주기억장치

- 메모리(휘발성)
- RAM을 매체로 사용하는 경우가 대부분

보조기억장치

- 마그네틱 디스크(비휘발성)

보조기억장치 용도

1. 전원이 나가도 유지할 파일 시스템용(file system)
2. 메모리 연장 공간인 스왑영역(swap area)
3. 운영체제는 당장 필요한 부분을 메모리에 놓고 나머지 부분을 디스크 스왑영역에 놓는다.(swap out)

저장장치의 계층 구조

1. primary
 - 레지스터 → 캐시 메모리 → 메모리
 - 용량이 적지만 필요한 것만을 올리면서 용량이 큰 저장장치를 가지고 있는 것과 마찬가지로 성능효과를 낼 수 있다.
2. secondary
 - 마그네틱 디스크 → 광디스크 → 마그네틱 테이프
 - 저장장치는 크고 느리지만 싸다.

하드웨어의 보안

운영체제는 여러 프로그램이 동시에 실행될 수 있는 다중 프로그래밍 환경에서 동작

커널모드

- 커널모드는 운영체제가 CPU의 제어권을 가지고 운영체제 코드를 실행하는 것이다.
- 중요정보에 접근해 위험한 상황을 초래할 수 있는 연산은 커널모드에서만 실행 되도록 한다.
- 일반 사용자 프로그램은 시스템에 위험한 연산을 하지 못하도록 통제한다.

사용자모드

- 일반 사용자 프로그램은 직접 위험한 명령 수행 불가
- 하지만 사용자 프로그램이 CPU를 가지고 있는 동안 중요한 연산을 수행해버리면 제어가 소용X
- 운영체제가 감시X
- 하드웨어적인 지원 필요
- CPU 내부에 모드비트를 두어 사용자 프로그램 감시

모드 비트(mode bit)

특권명령은 모드비트가 0일 때만, 즉 커널모드에서 운영체제에 의해서만 수행 가능

- 모드비트가 0이면 커널모드로서 모든 명령 수행 가능
- 모드비트가 1이면 사용자모드로서 제한된 명령만 수행 가능
- 운영체제가 사용자 프로그램에게 CPU 제어권을 넘길 때 모드비트를 1로 세팅
- 하드웨어 접근 등 보안이 필요한 중요한 명령을 수행해야할 경우 시스템 콜을 통해 운영체제가 대신 수행
- 인터럽트가 발생할 때 모드비트는 자동으로 0이 세팅
- 운영체제가 수행을 끝내고 다시 제어권을 넘겨줄 때 모드비트를 1로 세팅

메모리 보안

인터럽트 벡터와 인터럽트 처리루틴이 있는 곳은 보안 필요

기준 레지스터(**basic register**) - 어떤 프로그램이 수행되는 동안 그 프로그램이 합법적으로 접근할 수 있는 메모리상의 가장 작은 주소 보관

- 메모리 위 프로그램의 시작 주소

한계 레지스터(**limit register**) - 그 프로그램이 기준 레지스터값부터 접근할 수 있는 메모리의 범위를 보관

- 메모리 위 프로그램의 길이

CPU 보호

특정 프로그램이 무한반복문을 수행하는 등 CPU의 사용권한을 독점하고 있다면 빼앗을 방법이 없다.

타이머(timer)

- 타이머는 정해진 시간이 지나면 인터럽트 발생시켜 운영체제가 CPU의 제어권을 획득할 수 있게함
- 타이머에 의해 발생하는 인터럽트 처리루틴의 역할은 CPU 제어권을 빼앗아 다른 프로그램에게 이양하는 것

시스템 콜을 이용한 입출력 수행

입출력 명령은 특권명령

- 사용자 프로그램이 디스크의 파일에 데이터를 쓰는 행위
- 디스크의 파일로부터 데이터를 읽어오는 행위
- 키보드로부터 입력을 받는 행위
- 수행 결과를 화면에 출력하는 행위

시스템 콜

- 입출력 명령은 운영체제 코드에 구현되어있음
- 사용자 프로그램은 운영체제에 시스템 콜이라는 서비스 대행 요청으로 입출력 수행
- 소프트웨어적인 인터럽트