2장 과제

1731017 김민경

1. 컴퓨터 시스템은 계층구조(layered architecture)를 가진다. 어떤 층이 있으며, 계층구조의 목적 혹은 장점은 무엇인가?

컴퓨터 시스템은 사용자, 응용 프로그램, 운영체제 그리고 하드웨어로 계층구조를 가진다. 이와 같은 계층구조는 다음과 같은 장점이 있다.

첫째, 사용자는 하드웨어에 대한 자세한 지식이 없어도 컴퓨터를 쉽게 다룰 수 있다. 계층의 독립성은 새로운 하드웨어를 설치하거나 하드웨어를 변경하는 것도 돕는다.

둘째, 응용 프로그램이 직접 하드웨어를 다루지 못하도록 차단하므로서 하드웨어 사용의 충돌을 막는다.

셋째, 하드웨어는 컴퓨터를 구성하는 기계적 장치이다. 만약 운영체제 없이 하드웨어와 사용자만 존재했다면 입출력 관리, 메모리 관리, 하드웨어 간 데이터 변경 등 모두 사용자가 직접 관리하고 운영해야 했으므로 역할과 부담이 컸을 것이다. 하지만 이와 같은 계층분리로 인해 사용자의 부담이 줄었다.

2. 24 비트 CPU를 사용하는 시스템에서 설치 가능한 메모리의 최대 크기는 얼마인가? 단위는 MB. GB 등으로 표기하라.

16MB.

3. 컨텍스트란 무엇이며, 운영체제가 컨텍스트를 다루는 이유는 무엇인가? 컨텍스트 스위칭이란 어떤 행위이며, 이때 왜 CPU 레지스터들을 저장하는가?

컨텍스트란 어떤 프로그램이 실행 중인 일체의 상황을 말한다. 이와 같은 일체의 상황은 CPU 내에 저장된 레지스터, 메모리, 스택 등에 존재한다.
만약 컴퓨터가 매번 하나의 프로세스(스레드)만 처리한다면 작업이 끝날 때 까지 다음 작업은 기다릴 수 밖에 없다. 이는 반응속도가 매우 느리고 사용하기 불편하다. 따라서 CPU가 작업을 바꿔가며 실행을 하게 되면 실시간으로 처리되는 효과가 있어서 처리 속도를 높일 수 있다. 따라서 컨텍스트 스위칭의 개념이 등장하였다. 컨텍스트 스위칭이란 현재 실행중인 프로그램의 컨텍스트를 저장하고 다른 프로그램을 이전 진행하던 부분에서 부터 재개하게 위해 그 프로그램의 저장된 컨텍스트를 CPU로 옮기는 방법이다.

예를 들어 프로세스의 경우, 대부분 정보는 레지스터에 저장되고 PCB로 관리되고 있다. 구체적으로 보면 현재 실행되고 있는 프로세스는 PCB에 정보가 저장된다. 그리고 다음에 실행할 프로세스의 PCB 정보를 읽어 레지스터에 로드하면 CPU가 프로세스를 연속적으로 수행할 수 있다. 이와 같은 컨텍스트 스위칭은 처리 속도를 높일 수 있지만 빈도수가 많아지면 오버헤드가 발생될 수 있다는 문제가 있다.

그리고 컨텍스트 스위칭 과정에서 일반적으로 CPU 레지스터 값을 PCB에 넣어주게 되는데 이와 같은 이유는 CPU 레지스터가 현재 프로세스에 대한 데이터를 가지고 있기 때문이다. 구체적인 CPU 레지스터의 종류와 개수는 커널의 종류에 따라 다르지만 예를 들어 PC (Program Counter) 레지스터의 경우 현재 실행중인 메모리 번지, IR (Instruction Register)의 경우 현재 실행 중인 명령어, 데이터 레지스터는 이전에 실행한 결과 값이나 현재 실행에 사용될 데이터 값들, SP (Stack Pointer)는 스택의 톱 주소가 저장되어있다.

4. 운영체제와 커널은 같은 용어가 아니다. 구체적으로 어떤 차이점을 가지고 사용해야 하는가?

운영체제는 둘 소프트웨어, 커널, 디바이스 드라이버로 구성되어 있는 소프트웨어이다. 커널은 간단히 말하면 운영체제의 핵심 부분으로, 프로그램이며 코드이다. 커널은 부팅 후에 메모리에 상주하면서 CPU. 메모리. 입출력 장치. 등 컴퓨터 자원을 직접 제어하고 관리하는 코드들의 집합이므로 사용자가 직접 접근할 수 없는 영역이다.

운영체제는 시스템의 자원을 관리하는 시스템 프로그램이다. 반면 커널은 운영체제의 중요한 프로그램(부분) 중에 하나이다. 커널은 시스템의 소프트웨어와 하드웨어 사이의 인터페이스 역할을 한다. 반면 운영체제는 사용자와 컴퓨터 간 인터페이스 역할을 한다. 운영체제는 배치나 다중 프로그래밍, 분산처리, 실시간 시스템 등으로 분류가 되는데 커널은 모놀리식 커널과 마이크로 커널로 분류된다. 커널은 메모리 관리, 프로세스 관리, 디스크 관리 등을 한다. 하지만 운영체제의 경우 이와 같은 것을 포함하여 시스템 보호에 대한 책임이나 보안과 같은 총체적인 관리를 한다.

결론적으로 운영체제는 중요한 소프트웨어이고 시스템은 운영체제 없이는 실행될 수 없다. 커널은 운영체제의 중요한 프로그램이며 커널 없이는 운영체제가 동작할 수 없다.

5. 운영체제 커널이 자신에게 접근할 수 있도록 만들어 준 2가지 인터페이스는 무엇이며, 그 용도는 무엇인가?

2가지 인터페이스는 System call (시스템 콜) 과 interrupt (인터럽트) 이다.

시스템 콜은 사용자 영역과 운영체제 영역 간 인터페이스 중 하나이다. 운영체제 내 커널에 대한 요청이 들어오면 ~~프로그램은~~ 레지스터는 사용자 모드에서 커널 모드로 바꾸고 시스템 콜을 한다. 여기서 커널 모드로 진입하게 되면 특권 명령어를 실행 할 수 있게 되어 모든 메모리 주소와 하드웨어 접근할 수 있게 된다. 시스템 호출이 끝나면 다시 사용자 모드로 변환 되어서 운영체제의 서비스를 API (응용 프로그램 인터페이스) 를 통해 유저 프로그램에 서비스를 전달한다. 시스템 콜을 통해 제공되는 서비스는 다음과 같다. 프로세스 생성 및 관리, 메인 메모리 관리, 파일 접근 및 파일 시스템 관리, 장치 관리, 보호, 네트워크에서 패킷 송수신 등이 있다.

인터럽트는 CPU 내 프로세서한테 현재 프로세스를 중단하는 신호를 보내고 미리 정해진 ISR (인터럽트 서비스 루틴) 을 처리할 것을 요청하는 것이다. 이것은 보통 하드웨어 장치나 소프트웨어 프로그램에서 발생한다. 예를 들어, 만약 워드 프로세서를 사용 중이고 키를 눌렀다면 프로그램은 반드시 이와 같은 입력 프로세스를 즉시 처리해야한다. 그리고 "hello" 라고 타이핑을 했다면 이것은 또한 각각의 글자를 화면에 표시해야 한다는 5개의 인터럽트 요청을 만들게 된다. 이와 같이 마우스 버튼을 누르거나 화면을 터치했을 때에도 CPU에 인터럽트를 보내게 되는 것이다. 구체적으로 인터럽트의 동작 과정을 보면, 하드웨어와 소프트웨어 인터럽트 모두 인터럽트 핸들러에 의해 프로세스가 처리되는데, 이 핸들러를 ISR (인터럽트 서비스루틴) 이라고 한다. ISR은 하드웨어 장치로 부터 인터럽트 신호를 받으면 유발되는 소프트웨어 프로세스이다. 이것은 이와 같은 인터럽트 요청을 다루고 CPU 에 전달하여 현재 활성화된 프로세스를 중단시키는 역할을 한다. 그리고 ISR이 종료되면 프로세스는 다시 재개된다. 위에서 사례로 들었던 키보드 입력 과정을 보면, 키가 눌릴 때마다 ISR은 입력 과정을 처리한다. 파일에서 키보드를 눌렀을 때 ISR은 CPU 에게 해당 키가 눌렸다는 것에 대한 정보를 전달한다. CPU는 이 정보를 실행되고 있는 워드 프로세서에게 전달한다. 그리고 키를 다시 떼었다면 ISR은 키를 떼었다는 이벤트를 처리한다. 그래서 최종적으로 프로그램에게 커서를 그만 움직이라는 신호를 전달하게 되는 것이다.

정리하면, 시스템 호출은 프로그램이 커널에서 서비스를 요청할 수 있게 하는 방법이다. 프로그램은 커널로 하여금 메모리나 하드웨어 장치와 같은 자원에 접근하도록 전달한다 즉 프로그램에 대한 서비스를 커널에 요청하는 것을 말한다.

반면에 인터럽트는 CPU가 특정 작업을 즉시 수행하도록 하는 작업이다. CPU에게 현재 작업 중인 프로그램을 일시중지 시키도록 하고 다른 일을 즉시 처리하도록 한다.