운영체제_연습문제 풀이

2019111981 컴퓨터공학과 여은동

Q1) Keeping in mind the various definitions of *operating system,* consider whether the operating system should include applications such as web browsers and mail programs. Argue both that it should and that it should not and support your answers.

운영체제가 웹 브라우저 및 메일 프로그램과 같은 응용 프로그램을 포함해야 하는지 여부를 고려해라. 포함하면 안 되는 이유를 설명하고, 그에 대한 의견을 지지해라.

A1) 사실 운영체제 내부에 자주 사용하는 응용 프로그램, 즉 user program을 포함시키면 그 프로그램을 사용할 때 성능적인 부분에서 훨씬 뛰어난 점이 있을 수 있다. Kernel 외부에서 실행되는 응용프로그램을 사용하려면 Call interface system을 거쳐서 OS에게 전달이 되고 OS에 의해 internal device 및 external device를 사용하는 절차를 거쳐야 하기 때문에 시간이 걸릴 수 있는 반면, 운영체제 내부에 user program이 포함되어 있다면 복잡한 절차를 거치지 않아도 된다. 그러나 운영체제 내부에 user program이 포함되면, 이점보다 불이익이 더 크게 발현된다.

OS 프로그램은 많은 일들을 처리한다. Process들을 처리하는 일부터 시작해서 Resource들도 관리해야 하며, 웹 브라우저 및 메일 프로그램과 같은 응용 프로그램들도 컨트롤 해야 한다. 응용프로그램을 사용하는 user들에게는 계획된 행복함을 줘야 하고, HDD와 같은 하드웨어들에겐 효율적인 속도를 유지시켜 줘야 한다. 그런데 OS 프로그램 안에 응용 프로그램도 포함시켜버리면 보안적인 점에서 취약점이 생길 수도 있고, 운영 체제 자체가 커져버려서 user들과 하드웨어 사이의 효율적인 측면의 균형을 유지하기가 힘들어진다. 따라서 user program과 OS program을 따로 구분하는 것이 현명한 선택이다.

Q2) Which of the following instructions should be privileged? a. Set value of timer. / b. Read the clock. / c. Clear memory. / d. Issue a trap instruction. / e. Turn off interrupts. / f. Modify entries in device-status table. / g. Switch from user to kernel mode. / h. Access I/O device.

다음 명령들 중 어떤 것이 권한을 가져야 하는가? a. 타이머 값 설정 / b. 시계 읽기 / c. 메모리 지우기 / d. 트랩 명령 내리기 / e. 인터럽트 끄기 / f. 장치 상태 테이블의 항목 수정하기 / g. 사용자에서 kernel 모드로 전환하기 / h. I/O 장치 액세스 하기

A2) a. 타이머 값 설정 / c. 메모리 지우기 / e. 인터럽트 끄기 / f. 장치 상태 테이블의 항목 수정하기 / h. I/O 장치 액세스 하기

이 5개의 명령들이 권한을 가져야 한다. 권한을 가져야 하는 명령이랑 그렇지 않는 명령이랑 구분하려면 다음의 명령들이 user mode에서 수행이 가능한지를 따져보면 구분할 수 있다. 시계를 읽고, 트랩 명령을 내리고, 인터럽트를 끄는(ex. 알람 끄기) 작업들은 user mode에서 수행이 가능하다. 마지막으로 user mode에서 kernel mode로 전환하려면 "user mode"에서 명령을 내려야 하기 때문에 user mode에서 수행해야 하는 명령임을 알 수 있다.

Q3) Some early computers protected the operating system by placing it in a memory partition that could not be modified by either the user job or the operating system itself. Describe two difficulties that you think could arise with such a scheme.

일부 초기 컴퓨터들은 운영체제를 보호하기 위해서 사용자 작업이나 운영 체제 자체가 수정할 수 없는 memory partition에 운영 체제를 배치했다. 이러한 방식으로 인해 발생될 수 있는 두 가지 문제점을 설명하라.

A3) 운영체제를 보호하는 차원에서 user job이나 운영체제 자체가 수정할 수 없는 메모리 파티션에 운영체제를 배치하는 것 까지의 의도는 이해가 된다. 하지만 운영체제를 보호하려고 이러한 선택을 하면 다른 곳에서 보안의 문제점이 발생할 것이다. User job이나 운영체제 중 어떠한 것도 메모리 파티션에 접근이 불가능하기 때문에, 운영체제는 사용자를 변경할 필요가 있다. 그렇다면 명령 모드 변경이 불가피 할 것인데, 이러한 과정에서 민감 정보인 비밀번호와 같은 정보를 보호하기는 힘들 것이다. 또한 user mode에서 실행되는 프로그램이 다른 사용자의 메모리에 접근을 시도하면 이는 곧 error로 간주되기 때문에, 필수적으로 수행되어야 하는 작업들이 잘 실행되지 않게 된다. 따라서 이러한 과정이 중첩되다 보면 나중에는 결국 정보를 저장하기 힘들어질 것이고, 프로그램도 실행하기가 어려워질 것이다.

Q4) What is the purpose of interrupts? How does an interrupt differ from a trap? Can traps be generated intentionally by a user program? If so, for what purpose?

인터럽트의 목적은 무엇인가? 인터럽트는 트랩과 어떤 점에서 차이점이 있나? 사용자 프로그램에서 의도적으로 트랩을 생성할 수 있는가? 만약 생성할 수 있다면 어떤 목적으로 생성할 수 있는가?

A4-1) 인터럽트의 목적은 무엇인가?

인터럽트는 하드웨어에서 생성된 중요한 event들을 CPU에게 알림으로써 신속하게 event에 관련된 문제들을 우선순위에 따라서 해결하기 위해 발생한다. 인터럽트에도 우선순위가 있기 때문에, 인터럽트를 CPU에게 알려주지 않는다면 컴퓨터는 "중요한 일을 우선적으로" 해결하지 못하고 비정상적으로 작동할 것이다. 예를 들어 노트북의 하드웨어에서 가장 중요한 것은 배터리이기 때문에, 만약 배터리가 얼마 남지 않았음에도 불구하고 이를 user에게 제대로 알려주지 않으면 노트북은 꺼져버릴 것이다.

A4-2) 인터럽트는 트랩과 어떤 점에서 차이점이 있나?

기본적으로 본 강의에서 인터럽트는 하드웨어에서 생성된 event들을 알리는 것이고, 트랩은 소프트웨어에서 생성된 event들을 알리는 것이다. CPU에게는 특별한 flip/flop이 부착되어 있는데, 이 flip/flop은 만약 "divisor를 저장하는 레지스터의 값이 0"이거나 "존재하지 않은 파일을 열려고 할 때" 스스로 인터럽트를 발생시킨다. 어떠한 문제가 발생했을 때 해결하기 위해서 이를 알리는 것에서 인터럽트와 동일한 목적을 가지고 있다.

A4-3,4) 사용자 프로그램에서 의도적으로 트랩을 생성할 수 있는가? 만약 생성할 수 있다면 어떤 목적으로 생성할 수 있는가?

User program에서도 충분히 의도적으로 트랩을 생성할 수 있다. 예를 들어 계산기를 켜고 "4를 0으로 나누는 계산을 의도적으로 실행"한다고 한다면 트랩이 발생할 것이다. 이처럼 트랩은 산술 오류를 잡는 데에도 사용될 수 있고, 운영 체제 루틴을 호출하는 데에도 이용될 수 있다.

Q5) Direct memory access is used for high-speed I/O devices in order to avoid increasing the CPU's execution load.

a. How does the CPU interface with the device to coordinate the transfer?

b. How does the CPU know when the memory operations are complete?

c. The CPU is allowed to execute other programs while the DMA controller is transferring data. Does this process interfere with the execution of the user programs? If so, describe what forms of interference are caused.

CPU의 실행 부하 증가를 방지하기 위해서 고속의 I/O 장치에 직접 메모리 액세스가 사용된다.

a. CPU와 장치의 인터페이스를 통해 전송을 조정하는 방법은 무엇인가?

b. CPU는 메모리 작업이 완료된 시점을 어떻게 알 수 있나?

c. DMA 컨트롤러가 데이터를 전송하는 동안 CPU는 다른 프로그램을 실행할 수 있다. 이 과정이 사용자 프로그램의 실행을 방해하는가? 만약 그렇다면 어떤 형태의 간섭이 발생하는지 설명하라.

A5-a) CPU와 장치의 인터페이스를 통해 전송을 조정하는 방법은 무엇인가?

CPU의 인터페이스에는 정보의 전달을 위한 bus가 연결되어 있고, 이 bus는 4개(data lines, special lines, address lines, control lines)의 32bit line들로 구성이 되어있다. 이 중 data lines에 연결되어 있는 buffer or data register에 CPU와 device의 전송 속도 차이에서 발생되는 문제를 해결할 수 있는 장치가 있다. 기본적으로 CPU는 전송 속도가 매우 빠르고, device의 전송 속도는 그에 비해 느리다. 이 문제점을 해결하기 위해 CPU의 내부에는 buffer가 존재한다. buffer의 역할은 두 장치의 데이터 접점 타이밍이 안 맞을 경우를 대비해서 데이터를 임시저장 하는 것이다.

또한 CPU는 device가 접근할 수 있는 Status Register에 값을 기록함으로써 DMA 동작을 시작하도록 유도한다. Device는 CPU로부터 이 명령을 수신 받으면 해당 문제를 해결하기 위해 작업을 실시하게 되는 것이다.

A5-b) CPU는 메모리 작업이 완료된 시점을 어떻게 알 수 있나?

메모리 작업이 완료되면, device는 Status register를 "1"로 표시하여 메모리 작업이 완료되었음을 CPU에게 알린다. 즉 CPU에게 interrupt를 통해 알려준다.

A5-c) DMA 컨트롤러가 데이터를 전송하는 동안 CPU는 다른 프로그램을 실행할 수 있다. 이 과정이 사용자 프로그램의 실행을 방해하는가? 만약 그렇다면 어떤 형태의 간섭이 발생하는지 설명하라.

CPU는 기본적으로 main memory를 가장 많이 사용한다. Instruction Fetch -> Decode -> Decide Address -> Fetch Data -> Execute -> Store Result 이 과정만을 반복하는데 그 중 Instruction Fetch, Fetch Data, 그리고 Store Result 부분에서는 CPU가 메모리를 사용해야 한다. 그러나 메모리는 한 번의 전송에 하나의 데이터만 옮길 수가 있기 때문에, 메모리를 사용해야 하는 DMA 컨트롤러는 CPU의 동작을 방해하지 않기 위해서 "Cycle Stealing" 이라는 방법을 사용한다. 이 Cycle Stealing 방식의 원리는 CPU가 메모리를 사용하지 않는 순간에 DMA 컨트롤러가 잽싸게 HDD에서 MM으로 데이터를 가져오는 것이다. 하지만, 이러한 방식을 사용하면 CPU 입장에서는 속도가 떨어질 수밖에 없다. 왜냐하면 DMA 컨트롤러와 CPU의 속도차이가 심하기 때문이다. DMA 컨트롤러가 한번 동작할 시간이면 CPU는 300번 이상을 동작할 수 있는 시간이기에, 이 Cycle Stealing 방식을 이용하면 CPU 입장에서는 간섭이 발생할 수밖에 없다.