주홍철

2019년 1월 2일부터 쓰는 나의 운영체제 정리

운영체제

카카오는 갓카오

운영체제의 의미

운영체제는 컴퓨터 시스템의 자원을 효율적으로 관리해주는 것

* 자원 : 프로세스, 파일, 메시지, 기억장치, 입출력 장치 등

1. 주어진 하드웨어를 가지고 실행중인 프로그램들에게 짧은 시간씩 CPU를 번갈아 할당하는데 그것들, 메모리 공간을 적절히 어떻게 분배를 할 것인가 하는 고민에 대한 해결이 담긴 것.
2. 컴퓨터 시스템을 편리하게 사용할 수 있는 환경을 제공한다. 동시 사용자 / 프로그램 들이 각각 독자적 컴퓨터에서 수행되는 것 같은 환상을 제공한다. 하드웨어를 직접 다루는 복잡한 부분을 운영체제가 대행한다.

운영체제의 분류

1. 단일 작업, 한 번에 하나의 작업만 처리 그리고 다중작업은 동시에 두 개 이상의 작업 처리를 하는 것으로 분류한다. 현재의 컴퓨터는 다중작업이 가능하다.
2. 단일사용자, 다중사용자를 분류하는 운영체제
3. 일괄처리, 작업요청의 일정량을 모아서 한꺼번에 처리하는 것과 시분할, 타임쉐어링, 일저시간단위로 분할하여 사용한다. 현재의 컴퓨터, 짧은 응답시간으로 interactive 한 방식을 갖는다.
4. 실시간, 리얼타임 OS가 있다. 정해진 시간안에 어떠한 일이 반드시 종료됨을 보장된다. 원자로, 공장 제어, 미사일 제어, 반도체 장비 등에 사용된다. 요새는 확장이 되어서 1) 경성 실시간 시스템, 2) 연성 실시간 시스템이 존재한다.

멀티태스킹, 프로그래밍, 타임쉐어링, 멀티 프로세스는 비슷하게 여러 프로그램이 메모리에 올라가 있음을 말하고 cpu의 시간을 분할하여 나누어 쓴다는 의미이다. 멀티프로세서는 cpu가 여러개 붙어 있는 것을 말한다.

운영체제의 하는 일

크게 아래의 3가지 입니다.

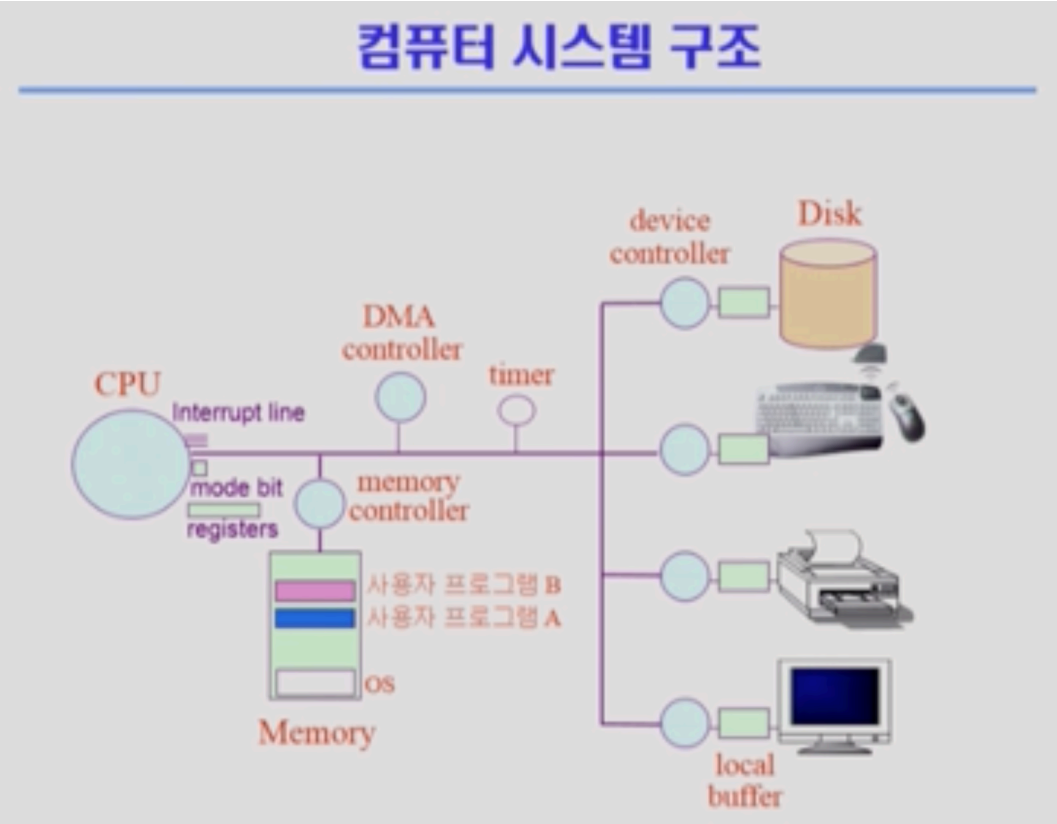
CPU 스케줄링, CPU를 누구에게 주는 가를 결정하는 것.

메모리관리, 한정된 메모리를 어떤 프로그램에 얼만큼 할당해야 하는가를 관리 하는 것입니다. 어떤 프로그램의 과거에 사용된 양을 기준으로 파악해서 메모리관리를 한다.

디스크 파일 관리, 디스크에 파일을 어떻게 보관하는지에 대한 것입니다.

I/O디바이스 관리, 입출력장치와 컴퓨터 간에 어떻게 정보를 주고 받게 할까에 대한 것입니다.
이외에도 프로세스관리, 프로세스의 생성과 삭제, 자원 할당 및 반환 등을 운영체제가 관리하고 합니다.

3강 - 컴퓨터 시스템 구조



출처:

컴퓨터 시스템 구조.

CPU는 “매클락사이클” 마다 메모리에서 기계어를 읽어서 실행한다.
컴퓨터와 연결되어있는 IO디바이스들은 작은 CPU가 붙어있는데 이걸 컨트롤러라고 하며 이 디바이스들의 작업공간을 로컬 버퍼라고 부른다.

CPU안에서 메모리보다 좀 더 빠른 데이터를 저장할 수 있는 공간이 있는데 이를 레지스트리라고 부른다. CPU안에 mode bit라는 것이 있다. 지금 실행하는 프로그램이 사용자 프로그램인지 운영체제인 지를 파악해주는 역할을 한다.

CPU는 직접적으로 IO디바이스에게 접근하지 않고 메모리에게만 접근해서 인스트럭션을 실행한다. 그러나 "디바이스의 디스크"에게 할 일이 생긴다면 메모리가 아닌 IO 디바이스의 컨트롤러, 장치를 제어하는 작은 CPU에 요청을 하게 된다. IO디바이스는 펌웨어가 있고 그 펌웨어가 "어떤 일"을 할 건지에 대한 정의가 있다.

요청을 한 후 IO작업시간동안 노는 것이아니다. 인스트럭션을 통해서 계속 일을 한다. 빠른 속도로 쉬지 않고 메모리에 접근하면서 일을 한다. 여러가지 일을 왔다 갔다 하면서 일을 하게 된다. 그러다가 IO디바이스의 일이 끝나면 CPU에게 인터럽트를 보낸다. 다른 일들을 하던 CPU의 권한은 운영체제로 다시 넘어가게 된다.
* 인스트럭션이란 CPU에서 메모리를 통해 기계어를 읽는 것을 말한다.

CPU는 보통의 경우 프로그램들에 각각 할당량을 주면서 효율적으로 작동하는데 "무한루프등 소모량이 많은 프로그램"이 작동할 때가 있다 그렇게 되면 효율적으로 작동을 못하게 되는데 방지하기 위해 타이머가 존재한다. 몇세컨드안에 실행되어야 한다라는 것을 정하고 특정 프로그램에 시간제한을 다는 역할을 한다.

CPU는 인스트럭션만 하는게 아니라 인스트럭션을 하면서 들어오는 인터럽트가 잇는지 계속 본다. 타이머가 인터럽트를 걸어오면 하던 프로그램을 중지하고 CPU의 사용권한이 사용자 프로그램으로부터 운영체제로 넘어가게 된다. 즉, CPU의 계속적인 작업을 위해 "타임쉐어링을 위해서" 타이머가 존재하는 것이다. 이를 타이머 인터럽트라고 한다.
- 그 프로그램으로부터 CPU를 뺐는다!

IO디바이스는 운영체제를 통해서만 작동하게끔해야 한다. 사용자프로그램은 나쁜 짓을 할 수 있기 때문에 운영체제를 통해서만 작동할 수 있어야 하고 이를 위해 modebit이 존재한다. modebit의 0, 1을 통해.
모니터모드 : os코드 실행할 수 있음.
사용자모드 : 사용자 프로그램 실행를 실행할 수 있음.
이를 구분하고 사용자모드일 경우에는 IO 디바이스에 시스템콜을 못하게 막는다.
- 한정된 인스트럭션만 가능하게끔 한다.
* 시스템콜, 사용자 프로그램이 운영체제의 서비스를 받기 위해서 커널 함수를 호출하는 것을 말한다.

메모리는 CPU만이 접근이 가능하고 하는 일도 많다. CPU는 메인 메모리 및 IO의 로컬 버퍼에게 접근 가능하다 보니까 CPU에게만 너무 많은 인터럽트요청이 들어온다. 그래서 DMA 컨트롤러를 두고 있다.
- 직접, 메모리를 접근하는 컨트롤러 direct memory access
이 때, CPU와 컨트롤러의 메모리 중첩 사용을 방지하기 위해 메모리 컨트롤러가 존재한다.

루틴이란.

입력, IO의 수행
1) 시스템 콜, 사용자 프로그램이 운영체제에게 I/O를 요청
2) trap을 사용하여 인터럽트 벡터의 특정위치로 이동
3) 제어권이 인터럽트 벡터가 가리키는 인터럽트 서비스 루틴으로 이동

4) 올바른 IO요청인지 확인 후 IO 수행
5) IO 완료 시 제어권을 시스템콜 다음 명령으로 옮김

인터럽트
- 하드웨어 인터럽트, 하드웨어가 발생시킨 인터럽트
- 트랩, 소프트웨어 인터럽트
1) Exception : 프로그램이 오류를 범한 경우
2) System call : 프로그램이 커널함수를 호출하는 경우

인터럽트 벡터, 해당 인터럽트의 처리 루틴 주소를 가지고 있음
인터럽트 처리 루틴, 해당 인터럽트를 처리하는 커널 함수

운영체제에 함수를 요청한다라고 하는 것.
- 사용자 프로그램이 소프트웨어적으로 인터럽트 라인을 설계해서 인터럽트를 거는 것을 말합니다.

```운영체제는 인터럽트에 의해 움직인다.

CPU스케쥴링이란.
CPU이용률을 극대화하기 위해, 즉 시스템의 자원을 효율적으로 사용하기 위해서 어느 프로세스가 CPU를 차지 할 것인가에 대한 순서 또는 방법을 결정짓는 작업을 말한다. 어느 시점에서 실행되고 있는 프로세스는 단 한개이며 여러개의 프로세스를 번갈아 가며 실행하는 것이다.

1. 생성, 프로세스는 준비단계에 머무른다. CPU를 점유하길 희망하는 상태준비 상태에서 실행 상태로 넘어가려면 작업 스케줄러가 선택해 주어야만 한다.
2. 디스패치, 준비 상태에서 실행 상태로 전이되는 과정을 말하며, 이는 작업 스케줄러가 해당 프로세스를 선택하여 실행되어지는 것으로, 이때 실행된 프로세스가 CPU를 점유하게 된다.
3. 인터럽트, 인터럽트 신호를 받게되면, 실행중이던 프로세스는 준비 상태로 전이되고, 우선순위(Priority)가 높은 프로세스를 실행 상태로 전이시킨다.

* 프로세스는 각각 우선순위를 부여받고, 우선순위에 따라 프로세스가 준비 상태로 전이되거나, 실행 상태로 전이된다.

4. 입출력 또는 이벤트 대기, CPU를 점유하고 있는 프로세스가 입출력 처리를 해야만 하는 상황이라면, 실행되고 있는 프로세스는 실행 상태에서 대기/보류 상태로 바뀝니다. 그리고 대기 상태로 바뀐 프로세스는 입출력 처리가 모두 끝날때까지 대기 상태로 머문다. 그리고 실행 상태이던 프로세스가 대기 상태로 전이됨과 함께, 준비 상태이던 또다른 프로세스가 실행 상태로 전이됩니다. 또한 대기 상태인 프로세스는 우선순위가 부여되지 않으며 스케줄러에 의해 선택될 수 없습니다.
입출력 또는 이벤트 완료, 입출력 처리가 끝난 프로세스는 대기 상태에서 준비 상태로 전이되어 스케줄러에게 선택될 수 있게 된다. 추가로 프로세스를 종료(Terminate)시킬 때에도 Blocked 상태를 거칠 수 있다.
```