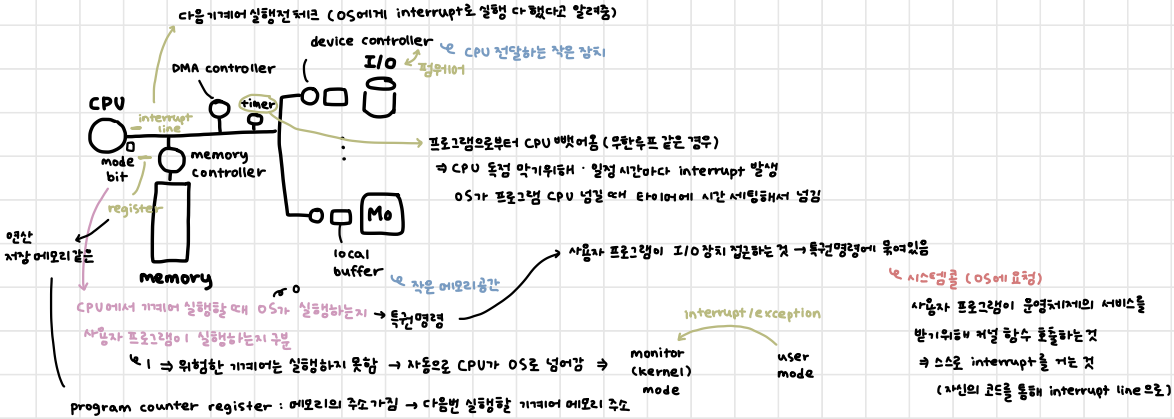
OS_ 컴퓨터시스템의 구조



Interrupt

하드웨어 인터럽트 : 하드웨어가 발생시킨 인터럽트

Trap (소프트웨어 인터럽트) — exception : 프로그램이 오류 범한 경우 / system call : 프로그램이 커널함수 호출한 경우

interrupt 마다 해야하는 동작이 다 다름

인터럽트 벡터 : 해당 인터럽트의 처리 루틴 주소를 가지고 있음

↳ 해당 인터럽트 처리하는 커널 함수

동기식 입출력 (synchronous I/O)

여럿이서 같이 무언가할 때 시간적으로 잘 맞게 일을 하는 것

일반적인 입출력 방식

: I/O 요청 후 입출력 작업이 완료된 후에야 제어가 사용자 프로그램에 넘어감

↳ 기다려야함 → CPU 낭비 ⇒ 다른 프로그램에 CPU 넘겨줌 ⊕ I/O도 낭비 (I/O도 한개만 진행가능)

↳ CPU가 I/O에 요청하고 I/O에서 일어나는 작업과 CPU에서 일어나는 작업이 싱크가 도는 것

비동기식 입출력 (ansynchronous I/O)

: I/O가 시작된 후 입출력 작업이 끝나기를 기다리지 않고 제어가 사용자 프로그램에 즉시 넘어감

↳ CPU가 I/O에 요청하고 I/O가 뭘하든 상관없이 다음 일을 진행

DMA (Direct Memory Access)

: 빠른 입출력 장치를 메모리에 가까운 속도로 처리하기 위해 사용 — interrupt 자주 발생하는 걸 막기 위해

↳ interrupt가 자주 일어나면 CPU에 비효율적 ⇒ CPU 외에도 메모리에 접근할 수 있는 장치를 하나 둔 것

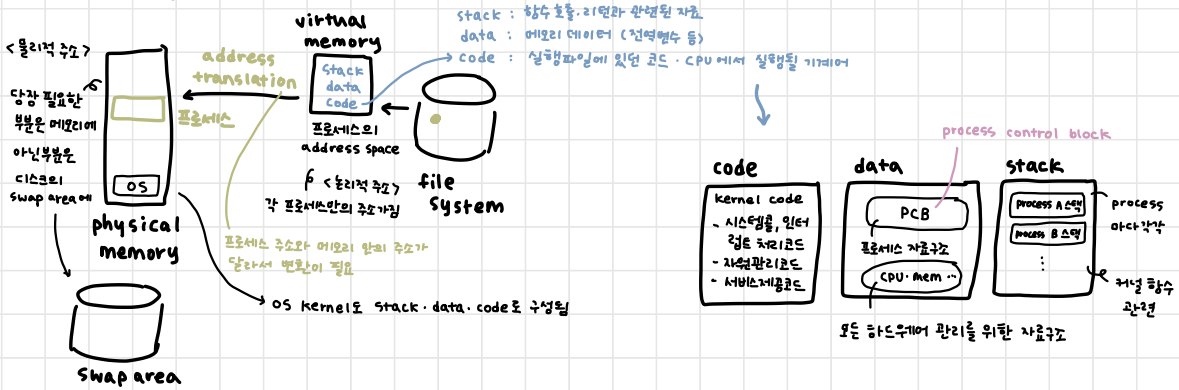
→ DMA가 직접 local buffer 에 있는 내용 memory에 복사 → interrupt 걸어서 작업이 끝났음을 CPU에게 알림

↳ block 단위

I/O를 수행하는 기계어 → I/O 전담 기계어 / 메모리 접근 기계어 따로

메모리 주소를 실제 메모리 뿐만 아니라 I/O 장치에도 주소 연장해 매겨서 I/O를 메모리 접근 기계어로 접근하는 방식도 있음

프로그램의 실행 (메모리 load)



함수 (function)

사용자 정의함수 / 라이브러리 함수 / 커널 함수 → 커널의 address space

프로세스 address space

커널의 함수로 이 함수 호출 → 시스템콜

프로그램 실행과정