프로세스

1. Job /program: 실행 할 프로그램 +데이터 실행 요청 전의 상태
2. Process: 실행을 위해 커널에 등록된 작업 커널(시스템)이 관리
   1. 실행중인 프로그램 프로세스에 대한 정보 관리
   2. Process control block 을 할당 받은 개체- 커널 공간 내에 존재

시스템 프로세스

사용자 프로세스

독립 프로세스 협력 프로세스

자원의 개념: 커널의 관리 하에 프로세스에게 할당 반납되는 수동적 개체

자원의 분류: H/w, S/W resources

Process Control Block: os가 프로세스 관리에 필요한 상태정보 저장

1. PID (process identification number)
2. 스케줄링 정보
3. 프로세스 상태

-프로세서 자원 간의 상호작용에 의해 결정

Created state- 작업을 커널에 등록 pcb할당 프로세스 생성

-가용 메모리 공간 체크 및 프로세스 상태 전이

Ready State- 프로세서 외에 다른 모든 자원을 할당 받은 상태

-Processor(CPU) 할당 대기 상태

-즉시 실행 가능 상태

-dispatch – ready state -> running state

Running State – 프로세서와 필요한 자원을 모두 할당 받은 상태

* Preemption(timer run-out) : running state -> ready state
* Block/sleep – running state -> asleep state (I/O 등 자원 할당 요청)
* Wake-up : asleep state -> ready state

Suspended State- 메모리를 할당 받지 못한 상태

* Memory image를 swap device에 보관

Terminated state- 프로세스 수행이 끝난 상태

* 자원 반납 커널 내에 일부 PCB 정보만 남아 있는 상태

1. 메모리 관리 정보
2. 입출력 상태 정보
3. Context save area 문맥 저장 영역
4. 계정정보

인터럽트: 예상치 못한 외부에서 발생한 이벤트 프로세스 중단

인터럽트 처리 : 1. 장소 원인 파악 결정 2. 인터럽 서비스

Context switching

1. Context- cpu register context : in CPU
2. Context saving-현재 프로세스 register context를 다른 메모리(pcb)에 저장하는 작업
3. Context restoring-register context를 프로세스로 복구
4. Context switching-context saving + context restoring 커널의 개입이 이루어짐

* Os의 큰 영향을 줌
* 스레드 사용 -> 불필요한 context switching을 줄이는게 중요