

질문	답변
프로세스가 무엇인가요?	프로세스는 실행 중인 프로그램 으로, 메모리에 적재되어 CPU 시간과 자원을 사용하여 작업을 수행하는 단위 입니다. 각각의 프로세스는 실행 중에 해당 프로그램의 코드, 메모리 할당, 입출력 작업 등을 담당하며, 운영체제에 의해 관리됩니다.
프로세스의 문맥이란 무엇인가요?	프로세스의 상태와 실행을 정확하게 복구 하는 데 필요한 정보입니다. CPU를 빼앗겼다가 다시 돌려 받을 때, 이전에 프로세스가 가지고 있던 상태를 정확히 알고 있어야만 제대로 된 복구가 됩니다. 정확한 재현을 위해 필요한 정보가 바로 프로세스의 문맥이다. 프로세스의 PC가 어떤 프로세스의 주소 공간 을 가리키고 있는지, stack 영역 에는 어떤 내용 들이 쌓여 있는지, 레지스터엔 어떤 값들이 존재 하는지 등등의 현재 상황을 규명하기 위한 다양한 정보들이 있습니다.
프로세스의 상태란 무엇인가요?	프로세스는 Running, Ready, Blocked 세 가지 상태를 가집니다. Running 상태는 CPU를 보유하고 명령을 실행하는 상태 입니다. 여러 프로세스가 수행되는 것처럼 보이지만 실제로는 한 번에 하나의 프로세스만 Running 상태에 있습니다. Ready 상태는 CPU를 할당받지 못해 대기하는 상태이며, Blocked 상태는 CPU를 할당받았지만 현재 명령을 실행할 수 없는 상태입니다 (주로 I/O 작업 중인 경우 : 프로세스가 파일을 읽거나 네트워크로 데이터를 받아야 할 때, 이 작업들은 시간이 걸릴 수 있습니다. 이때 해당 작업이 완료되기를 기다리는 동안 프로세스는 Blocked 상태로 들어가게 됩니다)
문맥 교환(Context Switching)이란 무엇이며 왜 필요한가요?	문맥 교환은 CPU가 한 프로세스에서 다른 프로세스로 전환 할 때, 현재 실행 중인 프로세스의 상태를 저장 하고 다음 실행할 프로세스의 상태를 복원 하는 과정을 말합니다. 이는 다중 프로세스 환경 에서 CPU를 효율적으로 공유 하기 위해 필요합니다.
문맥 교환이 발생하는 상황을 설명해주세요.	각 프로세스는 CPU를 일정 시간만 사용할 수 있습니다. 그 시간이 지나면 타이머 인터럽트가 발생하고, 운영체제는 현재 프로세스를 멈추고 다음 프로세스에 CPU를 할당하는 경우가 있습니다. I/O 같이 오랜 시간이 걸리는 작업은 프로세스가 CPU를 사용하는 동안 끝내지 못합니다. 이 경우 운영체제는 프로세스를 일시적으로 중단하고, I/O 요청을 처리하고 있는 동안 다른 프로세스에 CPU를 할당하며 문맥 교환이 이루어지게 됩니다,
문맥 교환(Context Switching)의 오버헤드는 무엇이며, 왜 발생하는가요?	문맥 교환은 시스템 성능에 영향을 미치는 오버헤드 중 하나입니다. 이는 CPU가 한 프로세스에서 다른 프로세스로 전환할 때, 현재 상태를 저장하고 복원해야 하는 작업으로, 이로 인해 시스템 자원과 시간이 소비 됩니다.

PCB(프로세스 제어 블록)이란 무엇인가요?	<p>PCB는 운영 체제에서 각 프로세스에 대한 중요한 정보를 저장하는 자료구조입니다. 각 프로세스는 고유한 PCB를 갖고 있으며, 프로세스가 CPU를 할당받아 작업을 수행하는 동안 상태를 추적하고 제어하는 데 사용됩니다.</p> <p>이는 프로세스의 상태, 레지스터 값, 메모리 관리 정보, 입출력 상태, CPU 스케줄링 정보 등을 포함합니다.</p> <p>예를 들어, 서버 애플리케이션에서는 PCB가 프로세스의 고유 식별자(PID)를 포함하고 있습니다. 이것은 각 프로세스를 고유하게 식별할 수 있도록 합니다.</p>
--------------------------	--------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------