

# OS 면접 질문

---

## 운영 체제란?

---

- OS란 무엇인지 설명해주세요.

사용자가 편하게 컴퓨터를 사용할 수 있도록 하드웨어와 시스템 자원을 관리하는 시스템이다.

---

## 프로세스와 스레드

---

- 프로세스와 스레드의 차이에 대해 설명하세요.

프로세스는 실행 중인 프로그램을 말하며, 완벽히 독립적이기 때문에 메모리 영역(Code, Data, Heap, Stack)을 다른 프로세스와 공유하지 않습니다. 프로세스는 최소 1개의 스레드(메인 스레드)를 가지고 있습니다.

스레드는 프로세스 내에서 Stack만 따로 할당 받고, 그 이외의 메모리 영역(Code, Data, Heap)을 공유하기 때문에 다른 스레드의 실행 결과를 즉시 확인할 수 있습니다. 스레드는 프로세스 내에 존재하며 프로세스가 할당받은 자원을 이용하여 실행됩니다.

- 멀티스레드의 개념과 장단점에 대해 설명해주세요.

멀티 스레드는 하나의 프로그램 실행에 여러 스레드를 구성하고 여러 스레드들이 공유 메모리를 통해 작업을 처리하도록 하는 방식입니다. 이를 통해 다수의 작업을 동시에 처리할 수 있고 메모리 자원을 효율적으로 활용할 수 있습니다.

그러나 공유 메모리를 사용하기 때문에 하나의 스레드가 데이터 공간을 망가뜨리면 모든 스레드가 작동 불능 상태가 되는 안전성 문제를 갖고 있습니다.

- 멀티 스레드의 동시성과 병렬성을 설명해주세요.

**동시성**은 멀티 작업을 위해 싱글 코어에서 여러 개의 스레드가 번갈아 실행하는 것을 말합니다. (동시에 실행하는 것처럼 보이지만 사실은 번갈아가며 실행하고 있는 것임)

**병렬성**은 멀티 작업을 위해 멀티 코어에서 한 개 이상의 스레드를 포함하는 각 코어들을 동시에 실행하는 것을 말합니다.

- 멀티 스레드 환경에서의 주의사항을 설명하세요.

다수의 스레드가 공유 데이터에 동시에 접근하는 경우에 상호배제 또는 동기화 기법을 통해 동시성 문제 또는 교착상태에 빠지지 않도록 주의해야 합니다.

- 멀티스레드의 안전성 문제를 보완하는 방법에 대해 말해주세요.

멀티스레드의 안전성 문제는 Critical Section 기법을 통해 보완할 수 있습니다. Critical Section 기법은 하나의 스레드가 공유 데이터 값을 변경하는 시점에 다른 스레드가 그 값을 읽으려고 할 때 발생하는 문제를 해결하기 위한 동기화 과정입니다.

- **멀티스레드 환경에서 싱글톤 패턴의 문제점**

서로 다른 스레드에서 거의 동시에 싱글톤 패턴으로 만든 객체 접근하면 객체가 두 번 생성되는 문제가 발생한다.

- **멀티스레드 환경에서 싱글톤 패턴의 문제점 해결법**

싱글톤 패턴으로 만들 객체의 생성을 동기화하여 특정 스레드에서 생성하면 다른 곳에서는 생성하지 않게 만들어야 한다.

- **멀티 프로세스와 멀티 스레드의 특징에 대해 설명하세요.**

멀티 프로세스는 하나의 프로세스가 죽어도 다른 프로세스에 영향을 끼치지 않고 계속 실행된다는 장점이 있지만 멀티 스레드보다 많은 메모리 공간과 CPU 시간을 차지한다는 단점이 있습니다.

멀티 스레드는 멀티 프로세스보다 적은 메모리 공간을 차지하고 문맥 전환이 빠르다는 장점이 있지만 하나의 스레드에 문제가 생기면 전체 스레드가 영향을 받으며 동기화 문제도 있다는 단점이 있습니다.

- **멀티 프로세스 대신에 멀티 스레드를 사용하는 이유를 설명해주세요.**

멀티 프로세스의 경우 프로세스를 생성하여 자원을 할당하는 시스템 콜을 할 때마다 오버헤드가 크지만, 멀티 스레드는 메모리 영역을 공유하기 때문에 시스템 자원을 효율적으로 사용할 수 있다

---

## 인터럽트(Interrupt)

- **인터럽트에 대해서 설명하세요.**

프로그램을 실행하는 도중에 예기치 않은 상황이 발생할 경우 현재 실행 중인 작업을 즉시 중단하고, 발생한 상황에 대한 우선 처리가 필요함을 CPU에게 알리는 것입니다.

---

## 시스템 콜(System Call)

- **fork()와 exec()에 대해서 설명해보세요.**

fork()와 exec()는 프로세스 생성과 관련된 시스템 호출(System Call)입니다.

fork()는 현재 실행 중인 프로세스를 복제하여 새로운 자식 프로세스를 생성하는 것을 말합니다. 이때, 자식 프로세스는 부모 프로세스의 상태를 그대로 상속받습니다. 즉, 자식 프로세스는 부모 프로세스와 동일한 코드, 데이터, 힙 영역을 공유하지만, 별도의 프로세스 ID(PID)를 가지게 됩니다.

exec()는 현재 프로세스의 메모리 공간에 새로운 프로그램을 덮어쓰는 것을 말합니다. 새로운 프로세스 이미지를 로드하고 실행하기 때문에 exec() 호출 이후에는 이전의 프로그램 코드, 데이터 및 자원은 모두 사라지고, 기존의 프로세스 이미지가 새로운 프로그램으로 대체됩니다. 이 때문에 동일한 PID를 유지하게 됩니다.

### PID

프로세스 식별자. UNIX 시스템에서는 PID는 프로세스에게 명령을 할 때 사용함.

---

# PCB와 Context Switching

---

- **PCB가 무엇이고 왜 필요한가요?**

PCB는 프로세스 정보를 담는 곳이고 앞으로 발생할 대기 중인 프로세스에 관한 정보를 저장하기 위해 필요하다.

- **컨텍스트 스위칭(Context Switching)이란?**

- 프로세스의 상태를 변경할 때 발생하는 작업
- 프로세서가 멀티 프로세싱을 구성, 어떤 하나의 프로세스를 실행하고 있는 상태에서 인터럽트 요청에 의해 다음 우선 순위의 프로세스가 실행되어야 할 때, CPU가 이전의 프로세스 상태를 PCB에 보관하고, 또 다른 새로운 프로세스의 정보를 PCB에 읽어 레지스터에 적재하는 과정입니다.

이 과정을 통해 다른 프로세스나 스레드 작업으로 전환하여 CPU의 사용률을 높일 수 있습니다. 단, Context Switching은 오버헤드가 발생하여 너무 빈번하게 발생하면 성능에 영향을 줄 수 있어 효율적인 스케줄링과 최적화가 필요합니다.

---

## IPC(Inter Process Communication)

---

- **IPC (Inter Process Communication)에 대해 설명해보세요. 그리고 IPC의 종류 중 하나를 설명해보세요.**

IPC란 '프로세스 간 통신'을 의미합니다. 다른 프로세스끼리 데이터를 주고 받고 상호작용하는 것을 의미합니다.

파이프(Pipe)는 IPC 종류중 하나입니다. 파이프는 한쪽 방향으로만 통신이 가능한 반이중 통신 방법입니다. 즉, 한 프로세스에서는 데이터를 쓰기만 가능하고, 다른 하나는 데이터를 읽기만 가능합니다.

---

## CPU 스케줄링 (Scheduling)

---

- **선점(preemption)과 비선점(non-preemption)이 무엇인지 설명해주세요.**

선점은 하나의 프로세스가 다른 프로세스 대신에 CPU를 차지할 수 있음을 말하고,

비선점은 하나의 프로세스가 끝나지 않으면 다른 프로세스는 CPU를 사용할 수 없음을 말합니다.

- **선점형 스케줄링과 비선점형 스케줄링의 차이**

선점형 스케줄링은 현재 실행 중인 프로세스가 다른 우선순위를 가진 프로세스에 의해 강제로 중단될 수 있는 방식입니다. 처리 시간이 긴 프로세스의 독점을 막을 수 있지만 잦은 Context Switching으로 인해 오버헤드가 커질 수 있습니다. 대표적인 예시로 Priority Scheduling, Round Robin 등이 있습니다.

비선점형 스케줄링은 현재 실행 중인 프로세스가 자발적으로 끝날 때까지 또는 입출력 작업 등의 이벤트가 발생할 때까지 CPU를 계속해서 점유하는 방식입니다. Context Switching으로 인한 오버헤드는 적지만 실행 중인 작업을 완료하는데 시간이 더 소요됩니다. 대표적인 예시로 FCFS, SJF, HRN 등이 있습니다.

- **콘보이 현상(convoy effect)**이란 무엇이고, 콘보이 현상이 발생할 수 있는 CPU 스케줄러 알고리즘은?

작업 시간이 긴 프로세스가 먼저 큐에 도착해서 다른 프로세스의 실행 시간이 전부 늦춰져 효율성을 떨어뜨리는 현상을 말합니다.

**FCFS**(First-Come First Served) 스케줄링은 순차적으로 먼저 큐에 들어온 작업부터 실행하므로 콘보이 현상이 발생할 수 있습니다.

---

## 데드락 (DeadLock, 교착 상태)

- 데드락에 대해 설명해주세요.

둘 이상의 프로세스들이 자원을 점유한 상태에서 서로 다른 프로세스가 점유하고 있는 자원을 요구하며 무한정 기다리는 상황을 말합니다.

예를 들어, 자원 A를 가진 프로세스 P1과 자원 B를 가진 프로세스 P2가 있을 때, P1은 B를 필요로 하고 P2는 A를 필요로 한다면 두 프로세스는 서로 자원을 얻기 위해 무한정 기다리게 됩니다.

- **교착상태의 4가지 조건은 무엇인가?**

교착 상태는 한 시스템 내에서 다음 네 가지 조건이 동시 성립될 때 발생한다.

1. 상호배제 : 한 자원에 대한 여러 프로세스의 동시 접근은 불가능하다
2. 점유와 대기 : 하나의 자원을 소유하고 다른 프로세스 또는 스레드의 자원을 요청하는 상태이다
3. 비선점(Non preemptive): 다른 프로세스에서 자원을 사용하는 동안 자원을 강제로 가져올 수 없다
4. 순환 대기: 각 프로세스가 다음 프로세스가 요구하는 자원을 가지고 있는 상태이다.

- 교착 상태의 해결 방법에 대해 말해보세요

1. 예방: 교착 상태의 4가지 조건 중 하나를 제거하여 해결하는 방법입니다.
2. 회피: 교착 상태 발생할 가능성을 피하면서 자원을 할당하는 방법입니다. (은행원 알고리즘)
3. 회복: 교착 상태를 일으킨 프로세스를 종료하거나, 할당된 자원을 해제시키는 방법입니다.
4. 무시: 회복과정의 성능저하가 심하면 아무런 조치도 취하지 않는 방법입니다. (비권장)

- **은행원 알고리즘이란?**

교착 상태(DeadLock)를 피하기 위한 회피 알고리즘.

프로세스가 자원을 요구할 때 시스템은 자원을 할당한 후에도 안정 상태로 남아있게 되는지를 **사전에 검사**하여 교착상태를 피합니다. 안정 상태에 있으면 자원을 할당하고, 그렇지 않다면 다른 프로세스들이 자원을 해지할 때까지 기다립니다.

---

## 경쟁 상태(Race Condition)

---

- **경쟁 상태(Race Condition)란 무엇인가요?**

컴퓨터 시스템에서 여러 프로세스나 스레드가 공유된 자원에 동시에 접근하거나 변경하려는 경우 발생할 수 있는 문제입니다.

---

## 세마포어(Semaphore)와 뮤텍스(Mutex)

---

- **세마포어와 뮤텍스의 차이를 설명해보세요.**

세마포어는 여러 개의 프로세스가 접근 가능한 공유 자원을 관리하는 방식입니다.

반면에 뮤텍스는 한번에 한개의 프로세스만 접근 가능하도록 관리하는 방식입니다.

이 때문에 세마포어는 다른 프로세스에 의해 해제될 수 있지만, 뮤텍스는 상호배제를 보장하며, 락을 획득한 프로세스만 락을 해제할 수 있습니다.

- **Critical Section(임계영역)에 대해 설명해주세요.**

- 임계 영역이란 프로세스간에 공유자원을 접근하는데 있어 문제가 발생하지 않도록 한번에 하나의 프로세스만 이용하게끔 보장해줘야 하는 영역을 말합니다.

- 여러 프로세스 또는 스레드가 공유 데이터를 동시에 접근하는 코드 영역을 의미합니다.

공유 데이터를 여러 프로세스가 동시에 접근할시에 예기치 못한 잘못된 결과를 만들 수 있기 때문에, 한 프로세스가 임계 구역을 수행할 때에는 다른 프로세스가 접근하지 못하도록 해야 합니다.

---

## 페이징과 세그먼테이션

---

- **페이징의 장점과 단점을 설명해주세요.**

장점: 메모리를 페이지 단위로 가져와서, 프로세스의 효율적인 운영이 가능하다.

단점: 페이지 크기별, 단위별로 페이지 폴트 현상이 발생할 수 있다.

- **메모리 관리 기법에 중 Paging에 대해 설명해보세요.**

Paging이란 논리주소의 메모리를 고정된 크기의 페이지(Page)로 나누어 관리하는 기법입니다.

페이지 테이블을 통해 논리 주소와 물리 주소 간의 매핑을 관리합니다. 페이지 테이블은 각 페이지에 대한 물리 주소 정보를 저장합니다. 페이지 단위로 데이터를 접근하므로 내부 단편화(Internal Fragmentation)가 발생하지만, 외부 단편화(External Fragmentation)는 발생하지 않습니다.

---

## 메인 메모리(main memory)

---

- **메모리 단편화란 무엇이고 해결 방법을 설명해주세요**

메모리의 공간이 작은 조각으로 나뉘어져 사용가능한 메모리가 충분히 존재하지만 사용이 불가능한 상태를 메모리 단편화가 발생했다고 합니다.

해결 방법에는 메모리 압축(디스크 조각 모음), 메모리 통합(단편화가 발생된 공간들을 하나로 통합시켜 큰 공간으로 만드는 기법) 이 있습니다.