

01.15

피드백 시간

인터페이스의 장점

- 클래스의 다중 구현 및 인터페이스 간 다중 상속이 가능합니다.
- ocp를 지키며 클래스 간 코드 결합도가 낮은 프로그래밍이 가능해집니다.
- 마커 인터페이스, 함수형 인터페이스로 이용 가능합니다.

뮤텝스 - 프로세스단위의 동기화방식

2가지 임계영역에 대한 lock 구현 방식 뮤텝스, 세마포어

- Mutex(Mutual Exclutsion)
 - 임계 구역에 대한 프로세스 간의 시간이 겹치지 않도록 하는 방식
 - 스레드 간의 경쟁을 기반으로 함
 - Lock을 소유한 프로세스만이 임계 영역에 접근 가능
 - 스레드 단위의 동기화 방식
- Semaphore
 - 시그널 매커니즘(임계 영역 전 wait(), 임계 영역 후 signal())을 통해, 프로세스 간 임계 영역을 사용하도록 한 방식
 - 2개 이상의 프로세스 간 경쟁을 기반으로 함
 - 시스템 범위에서 커널이 소유한 세마포어를 통해 임계 영역에 접근 가능
 - 기다리는 방식 - Busy Waiting(초기) → Block-Wakeup(현재)

http

Http 1.0

- 1996
- HEAD, POST 추가

- http 헤더, 바디, 응답코드 추가됨.
- **비지속 연결**(Non-Persistent Connection) - TCP 세션 유지 X
 - 1GET - 1CONNECTION
- 단순히 open/operation/close을 통한 flow의 제한 → 네트워크 혼잡, disconnect 발생
→ 서버에 재접속 시도 → 서버 과부하, 성능 저

Http 1.1

- 1999
- OPTION, PUT, DELETE, TRACE 추가
- **지속 연결**(Persistent Connection, Keep-alive) - TCP 세션 유지 - O
 - N GET - 1CONNECTION
 - 파이프 라이닝 - 커넥션 내 순차 요청-응답
 - Network Latency 줄임
 - HOL 문제
- Host Header - 버추얼 호스팅(1IP - N DOMAIN)
- 강력한 인증 절차 → 프록시에서 사용자의 인증을 요구
 - proxy-authentication, proxy-authorization 헤더 추가