

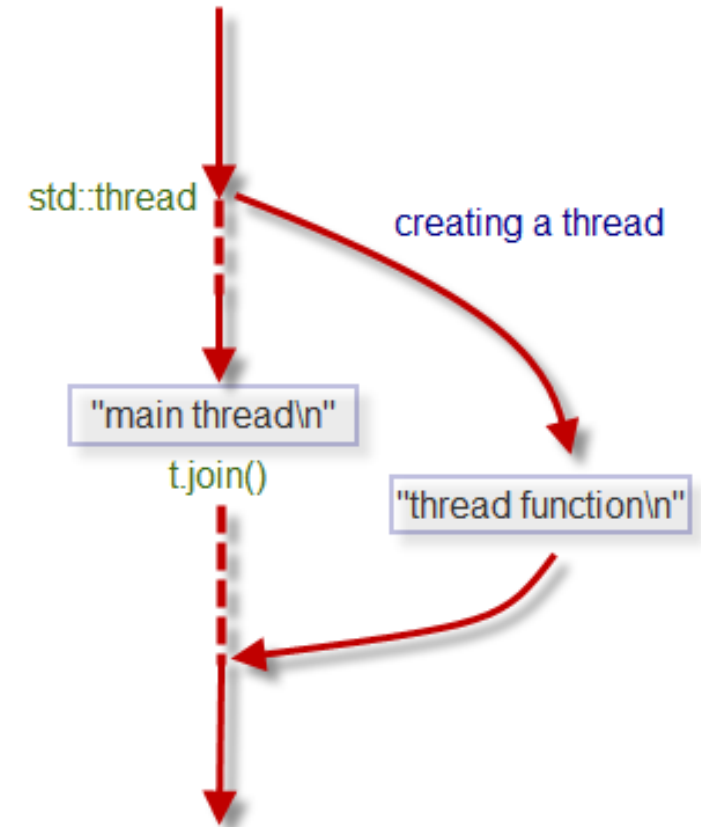
CODINGO x **posco**

K-Digital Training 스마트 팩토리 3기

Thread

Thread

- 어떠한 프로그램 내에서, 특히 프로세스 내에서 실행되는 흐름의 단위
- 일반적으로 한 프로그램은 하나의 스레드를 가지고 있지만, 프로그램 환경에 따라 둘 이상의 스레드 동시에 실행가능
- 이러한 실행 방식을 **멀티스레드 (multithread)** 라고 한다.



Thread Scheduling

- 다중 스레드 환경에서 어떤 스레드가 CPU 를 할당 받고 실행될지를 관리하는 프로세스.
- 스레드 스케줄링은 운영 체제에 따라 다를 수 있음

Thread Scheduling

- 주요 개념과 원칙

1. 스레드 우선순위

- 스레드가 CPU를 얼마나 자주 할당 받을지 결정하는데 사용

2. 스케줄링 알고리즘

- 라운드 로빈, 우선 순위 기반 스케줄링, 우선 순위 봉쇄를 방지하기 위한 스케줄링

3. 문맥 교환

- CPU에서 실행 중인 스레드를 바꿀 때 문맥 교환을 수행
- 실행 중인 스레드의 상태를 저장하고 다음 실행할 스레드의 상태를 복원

4. 우선순위 역전

- 높은 우선순위를 가진 스레드가 낮은 우선순위를 가진 스레드에게 의존하는 경우 발생

Race condition

- 다중 스레드 또는 다중 프로세스 환경에서 여러 스레드 또는 프로세스가 공유된 자원에 동시에 접근하려고 할 때 발생하는 프로그래밍 문제

Race condition

- 주요 특징과 원인
 1. 공유 자원
 2. 동시 접근
 3. 올바른 순서 없음
- 그로 인한 문제
 1. 데이터 손실 또는 오염
 2. 데드락(deadlock)
 3. 비정상적인 동작

Race condition

- 해결방법

- **std::mutex**

- 여러 스레드가 동시에 공유 데이터에 접근하는 것을 막기 위해 사용
 - lock, unlock

- **std::atomic**

- 원자적 연산을 수행하기 위한 도구, 공유 변수를 안전하게 업데이트
 - 여러 스레드 간에 동시에 변수 값을 수정하더라도 데이터 무결성을 보장

- **std::condition_variable**

- 주로 스레드가 특정 조건을 만족할 때까지 대기하고, 조건이 충족되면 다른 스레드에게 신호를 보내고 깨우는데 사용
 - 일반적으로 mutex와 함께 사용