



1

# 스레드의 개념

## 1 개념

- ▶ 프로세스의 특성인 자원과 제어에서 제어만 분리한 실행 단위
- ▶ 프로세스 하나는 스레드 한 개 이상으로 나눌 수 있음
- ▶ 프로세스의 직접 실행 정보를 제외한 나머지 프로세스 관리 정보 공유
- ▶ 다른 프로시저 호출, 다른 실행 기록(별도 스택 필요)
- ▶ 관련 자원과 함께 메모리 공유 가능하므로 손상된 데이터나 스레드의 이상 동작 고려

## 1 개념

### 경량 프로세스 LWP, Light Weight Process

- 프로세스의 속성 중 일부가 들어 있는 것

### 중량 프로세스 HWP, Heavy Weight Process

- 스레드 하나에 프로세스 하나인 전통적인 경우

## 1 개념

- ▶ 같은 프로세스의 스레드들은 동일한 주소 공간 공유
- ▶ 동일 프로세스의 여러 스레드는 코드 영역, 데이터 영역, 운영체제 자원(예: 열린 파일, 시그널)을 공유
- ▶ 프로세스가 다수의 제어 스레드를 가진다면, 프로세스는 동시에 하나 이상의 작업을 실행할 수 있음

# 스레드의 개념

## 스레드 병렬 수행

- ▶ 프로세스 하나에 포함된 스레드들은 공동의 목적 달성을 위해 병렬 수행
- ▶ 프로세스가 하나인 서로 다른 프로세서에서 프로그램의 다른 부분 동시 실행

## 3 스레드 병렬 수행의 이점

- ▶ 사용자 응답성 증가
- ▶ 프로세스의 자원과 메모리 공유 가능
- ▶ 경제성 좋음
- ▶ 다중 처리(멀티 프로세싱)로 성능과 효율 향상

## 2 단일 스레드와 다중 스레드

## 2 단일 스레드와 다중 스레드

### 1 단일 스레드와 다중 스레드

#### 단일 스레드

- ◆ 프로세스 하나에 스레드 하나가 실행되는 전통적인 방식으로 스레드의 개념이 불확실함

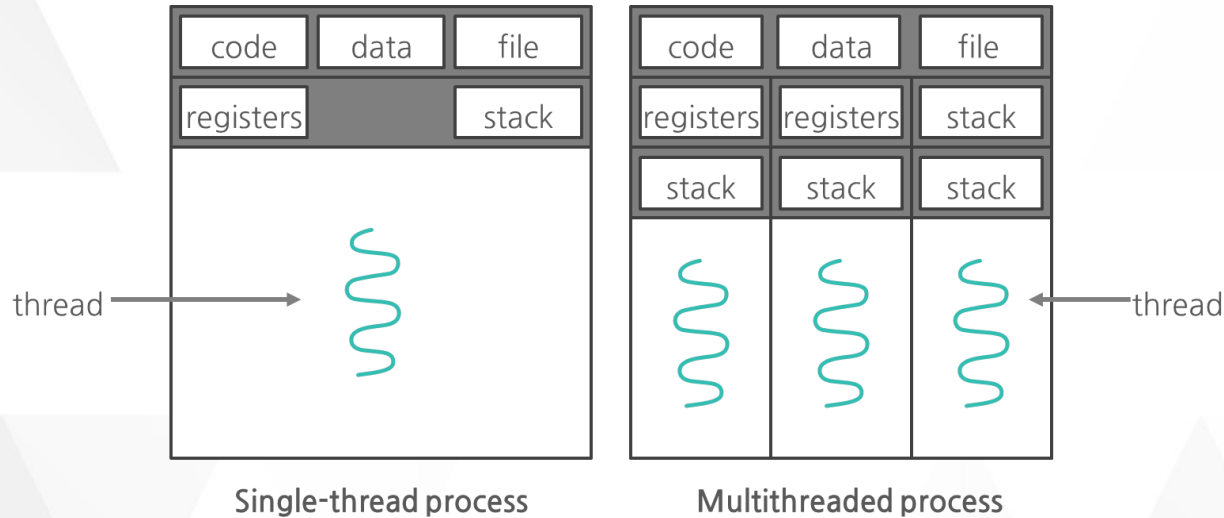
#### 다중 스레드

- ◆ 프로그램 하나를 여러 실행 단위로 쪼개어 실행한다는 측면에서 다중 처리(다중 프로세싱)와 의미가 비슷함



## 2 단일 스레드와 다중 스레드

### 1 단일 스레드와 다중 스레드



- ◆ 스레드별로 실행 환경 정보가 따로 있지만 서로 많이 공유하므로, 프로세스보다 동일한 프로세스의 스레드에 프로세서를 할당하거나 스레드 간의 문맥 교환이 훨씬 경제적임

## 2 단일 스레드와 다중 스레드

### 2 스레드 이용 예제

- ▶ 현대 운영체제의 대다수 소프트웨어 패키지는 다중 스레드를 지원함
- ▶ 웹 브라우저의 스레드 이용 예
  - 이미지와 텍스트를 표시(display)하는 스레드 1개
  - 네트워크로부터 데이터를 읽어오는 스레드 1개
- ▶ 워드 프로세서의 스레드 이용 예
  - 그래프를 표시(Display)하는 스레드 1개
  - 스펠링, 문법 검사를 수행하는 스레드 1개

## 2 단일 스레드와 다중 스레드

### 2 스레드 이용 예제

#### ◆ 웹 서버

- 단일 스레드 웹 서버
  - 한 클라이언트 요청에 대한 서비스를 완료할 때까지 다른 클라이언트 요청에 대한 서비스를 기다려야 함
- 다중 프로세스 웹 서버
  - 스레드가 일반화되기 전에 사용되었으며, 새로운 프로세스를 생성하여 추가적인 클라이언트 요청을 처리함
- 다중 스레드 웹 서버
  - 프로세스에 비해 오버헤드가 작은 스레드를 생성하여 여러 클라이언트에 대해 동시에 서비스함

## 2 단일 스레드와 다중 스레드

### 2 스레드 이용 예제

- ▶ 많은 운영체제 커널들이 현재 다중 스레드 기반
  - 여러 스레드들이 커널 내에서 동작함
  - 각 스레드는 장치를 관리하거나 인터럽트를 처리하는 등 특정 작업을 수행함

## 2 단일 스레드와 다중 스레드

### 3 다중 스레드 프로그램의 장점

#### 반응성(Responsiveness)

- 상호작용 응용의 다중 스레딩은 하나의 스레드가 일시 봉쇄(Blocking)되어도 다른 스레드는 수행을 계속하게 함 ⇨ 사용자 응답성 증가

#### 자원공유(Resource Sharing)

- 하나의 프로세스에 속한 여러 스레드는 메모리 등의 자원을 공유

## 2 단일 스레드와 다중 스레드

### 3 다중 스레드 프로그램의 장점

#### 경제성(Economy)

- 스레드는 자신이 속한 프로세스의 자원들을 공유하기 때문에 스레드를 생성하고 문맥을 교환하는 것이 보다 더 경제적임

#### 확장성(scalability) - 다중 프로세서 구조의 활용

- 다중 프로세서(다중 코어) 시스템에서 스레드는 서로 다른 프로세서(코어)에서 병렬로 실행될 수 있음

## 2 단일 스레드와 다중 스레드

### 4 다중 스레드 모델

#### ▶ 사용자 스레드(User thread) vs 커널 스레드(Kernel thread)

##### 사용자 스레드

- ❖ 사용자 수준(User level)에서 지원

##### 커널 스레드

- ❖ 커널 수준(Kernel level)에서 지원

- 운영체제 커널은 커널 수준에서 스레드를 지원하고 관리함
- 사용자 스레드는 프로그래머에게는 보이나 커널에게는 알려지지 않는 스레드임

## 2 단일 스레드와 다중 스레드

### 4 다중 스레드 모델

- ▶ 일반적으로 사용자 스레드는 커널 스레드보다 더 빨리 생성하고 관리할 수 있음, 왜냐하면 커널의 개입이 필요하지 않기 때문임



## 2 단일 스레드와 다중 스레드

### 4 다중 스레드 모델

#### 사용자 스레드(User thread)

- ◆ 스레드 관리는 사용자 레벨 스레드 라이브러리에 의해 이루어짐
- ◆ 사용자 스레드는 커널 위(Above)에서 동작하므로, 커널의 지원 없이 관리됨
- ◆ 세 가지 기본 스레드 라이브러리
  - POSIX Pthreads
  - Win32 threads
  - Java threads

## 2 단일 스레드와 다중 스레드

### 4 다중 스레드 모델

#### 커널 스레드(User thread)

- ◆ 커널 스레드는 운영체제 커널에 의해 작업 지원받고 관리됨
- ◆ 커널 스레드 예제
  - Windows XP/2000
  - Solaris
  - Linux
  - Tru64 UNIX
  - Mac OS X

## 2 단일 스레드와 다중 스레드

### 4 다중 스레드 모델

▶ 사용자 스레드들과 커널 스레드들 사이의 관계

다대일 (Many-to-One) 모델

일대일 (One-to-One) 모델

다대다 (Many-to-Many) 모델

## 2 단일 스레드와 다중 스레드

### 4 다중 스레드 모델

#### 다대일 모델

- ◆ 여러 사용자 스레드들이 하나의 커널 스레드에 매핑됨
  - 스레드 관리는 사용자 공간의 스레드 라이브러리에 의해 이루어짐
  - 사용자가 원하는 만큼의 많은 사용자 스레드를 생성할 수 있으므로 효율적임
  - 하나의 스레드가 봉쇄형(blocking) 시스템 호출을 할 경우, 전체 프로세스가 봉쇄됨
  - 한 번에 하나의 스레드만이 커널에 접근할 수 있기 때문에, 다중 스레드가 다중 프로세서에서 돌아도 병렬로 작동할 수 없음

## 2 단일 스레드와 다중 스레드

### 4 다중 스레드 모델

#### 다대일 모델

##### ◆ 다대일 모델 예제

- 솔라리스 스레드 라이브러리
- GNU Portable 스레드

## 2 단일 스레드와 다중 스레드

### 4 다중 스레드 모델

#### 일대일 모델

◆ 하나의 사용자 스레드는 하나의 커널 스레드에 매핑됨

- 다대일 모델보다 더 많은 병렬성을 제공함
- 하나의 스레드가 봉쇄되더라도 다른 스레드들은 계속 수행됨
- 다중 처리기(혹은 다중 코어)에서 다중 스레드가 병렬적으로 수행되기도 함
- 사용자 스레드 생성은 대응하는 커널 스레드를 필요로 함

## 2 단일 스레드와 다중 스레드

### 4 다중 스레드 모델

#### 일대다 모델

- ▶ 하나의 사용자 스레드는 하나의 커널 스레드에 매핑됨
  - 하나의 프로세스가 생성할 수 있는 스레드 수는 다대일 모델보다 적음
    - ⇒ 너무 많은 스레드를 생성하지 않도록 주의해야 함
- ▶ 일대일 모델 예제
  - Windows NT/XP/2000
  - Linux
  - Solaris 9과 그 이후 버전

## 2 단일 스레드와 다중 스레드

### 4 다중 스레드 모델

#### 다대다 모델

- ◆ 여러 사용자 스레드가 여러 커널 스레드와 매핑됨
  - 정확히는 여러 사용자 스레드들이 그 보다 더 적은 수의 커널 스레드와 매핑됨
  - 운영체제가 충분한 수의 커널 스레드를 생성하도록 함
  - 응용은 원하는 수 만큼의 사용자 스레드들을 생성하고 이들에 대응하는 커널 스레드들이 다중 처리기에서 병렬로 수행될 수 있음
  - 다대일과 일대일의 장점을 취한 접근법



## 2 단일 스레드와 다중 스레드

### 4 다중 스레드 모델

#### 다대다 모델

##### ◆ 다대다 모델 예제

- Solaris 9 이전 버전
- ThreadFiber 패키지를 가진 Windows NT/2000

# 3 스레드의 상태변화

## 1 스레드의 상태변화

- ▶ 프로세서 함께 사용, 항상 하나만 실행
- ▶ 한 프로세스에 있는 스레드는 순차적 실행, 해당 스레드의 정보 저장레지스터, 스택)
- ▶ 프로세스 생성하면 해당 프로세스의 스레드도 함께 생성됨
  - 단, 스레드 생성에서는 운영체제가 부모 프로세스와 공유할 자원 초기화 필요 없음
- ▶ 프로세스의 생성과 종료보다는 오버헤드 훨씬 적음
- ▶ 스레드 한 개가 대기 상태로 변환 시 전체 프로세스 대기 상태로 변환하지 않음

### 1 스레드의 상태변화

- ▶ 실행 상태의 스레드가 대기 상태가 되면 다른 스레드 실행 가능(서로 독립적이지는 않음)
- ▶ 프로세스 하나에 있는 전체 스레드는 프로세스의 모든 주소에 접근 가능하여 스레드 한개가 다른 스레드의 스택 읽거나 덮어 쓰기 가능
- ▶ 프로세서는 여러 사용자가 생성하여 서로 경쟁적으로 자원 요구하고 서로 다른 관계를 유지해야 하지만, 스레드는 사용자 한 명이 여러 스레드로 개인 프로세스 하나 소유

- ▶ 정보 저장
- ▶ 프로세스 제어 블록은 스레드 제어 블록의 리스트
- ▶ 스레드 간에 보호 하지 않음

### ▶ TCB의 내용

- 실행 상태  
: 프로세서 레지스터, 프로그램 카운터, 스택 포인터
- 스케줄링 정보  
: 상태(실행, 준비, 대기), 우선순위, 프로세서 시간
- 계정 정보