

Slide 0

2부 프로세스 관리 (5장. 스레드)

개요

- 스레드란?
 - 하나의 실행 흐름으로서 작업 단위
 - 스레드 (LWP: lightweight process) ↔ 일반 프로세스 (heavyweight process)
 - 전통적 프로세스는 하나의 스레드를 가짐
 - 하나의 프로세스가 여러개의 스레드로 구성되면 동시에 하나 이상의 작업이 가능

Slide 1

- 구성
 - * 스레드 ID
 - * PC (Program Counter)
 - * 레지스터 집합
 - * 스택
- 하나의 프로세스내의 스레드는 다른 스레드와 다음을 공유
 - * 주소공간
 - * 텍스트
 - * 데이터
 - * 자원 (파일, 신호등등)

개요 (계속)

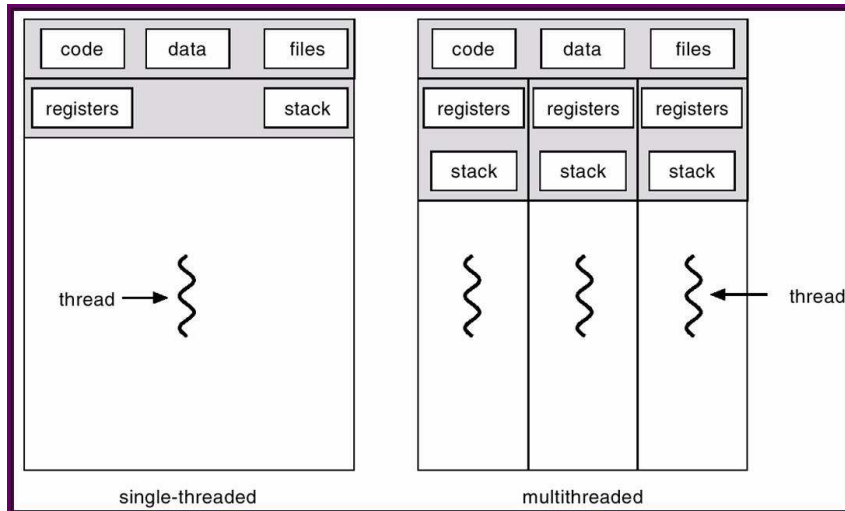
Slide 2

- 다중 스레드 효과
 - 좋은 예)
 - ➔ 웹서버: 요청별로 스레드가 처리
 - ➔ 웹브라우저: 네트워크에서 데이터를 받는 스레드와 받은 데이터를 처리하는 스레드로 나눔
 - ➔ 워드프로세서
 - 키보드 입력을 받아들이는 스레드
 - 화면에 표시하는 스레드
 - 철자나 문법을 검사하는 스레드
 - 나쁜 예)
 - ➔ 단일 프로세서 환경에서 계산 중심 프로그램

개요 (계속)

Slide 3

- 단일 및 다중 스레드 프로세스



개요 (계속)

- Slide 4**
- 동기
 - 한 프로세스에서 여러개의 작업을 동시에 수행할 수 있음
 - 새로운 프로세스를 생성하여 처리하는 것 보다 빠르고 비용이 적음
 - 유닉스의 경우 프로세스보다 스레드 생성이 10배 정도 빠름
 - 예) 웹서버가 하나의 프로세스로 구성된 경우
 - * 서버 프로세스가 요청을 처리 →한 순간에 하나의 요청만 처리가능
 - * 서버 프로세스가 요청을 받았을때 새로운 프로세스를 생성하여 작업을 맡김 (예전 방법으로 비효율적)
 - ↳ 서버 프로세스가 요청을 받았을때 새로운 스레드를 생성하여 작업을 맡김

개요 (계속)

- Slide 5**
- 장점
 - 응답성 (responsiveness)
 - 다중 스레드 프로그램의 일부가 block 되어도 나머지 부분은 계속 수행 됨
 - 자원공유 (resource shareing)
 - 한 프로세스내에서 메모리 및 자원을 공유하므로 자원 공유 및 통신이 용이
 - 경제성 (economy)
 - * 프로세스 생성보다 스레드 생성이 비용이 적게 듦, 문맥전환이 용이
 - * 예) 솔라리스에서
 - ↳프로세서 생성이 스레드 생성보다 30배 느리고
 - ↳문맥전환하는데 5배 느림
 - 멀티프로세서 구조에서 유용
 - * 멀티프로세서에서 각 스레드가 각 프로세서에 할당되어 수행 가능
 - concurrency 향상됨
 - * 하나의 스레드로된 프로세스는 여러 프로세서가 있어도 한 프로세서에서 만 실행 됨

개요 (계속)

Slide 6

- 사용자와 커널 스레드
 - 스레드를 사용자 공간이나 커널공간 혹은 두 공간 모두에서 구현 가능
 - 사용자 스레드
 - * 설명
 - 스레드를 사용자 공간에서 구현
 - 스레드 라이브러리를 통해 제공됨
 - 커널 지원없이 스레드 생성, 스케줄링, 관리를 제공
 - 커널은 전혀 관여하지 않음
 - 커널은 스레드 단위로 스케줄하지 않고 프로세스 단위로 스케줄
 - * 장점
 - 모드변경이 필요없어 스레드의 생성과 관리가 빠름
 - 프로세스마다 자체적인 스케줄링이 가능
 - * 단점
 - 커널이 프로세스 단위로 스케줄함으로 하나의 사용자 스레드가 block 되면 다른 스레드도 block 됨
 - * 예) POSIX 의 Pthread, Mach 의 C-thread, Solaris 의 UI-thread
 - * 운영체제가 스레드를 지원하지 않을 경우 스레드를 지원하는 유일한 방법

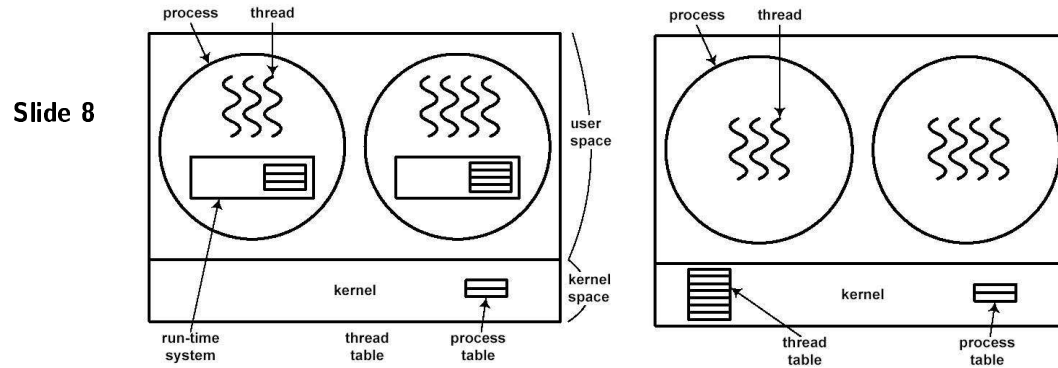
개요 (계속)

Slide 7

- 커널 스레드
 - * 설명
 - 스레드를 커널 공간에서 구현
 - 커널이 스레드의 생성, 스케줄링, 관리 담당
 - * 장점
 - 하나의 스레드가 block 되어도 다른 스레드는 계속 수행됨
 - * 단점
 - 모드변경이 필요해서 스레드의 생성 관리가 사용자스레드보다 느림
 - * 예) 윈도우 NT, 2000, Solaris 2, Digital UNIX는 커널스레드 지원

개요 (계속)

- 사용자 스레드와 커널 스레드



다중 스레드 모델

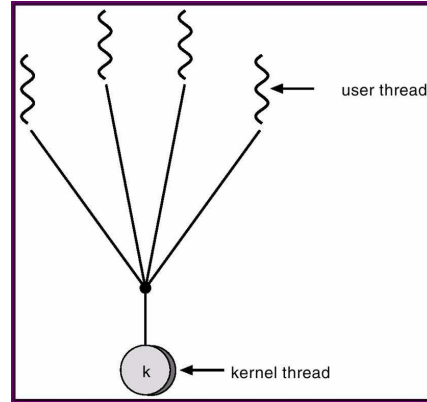
- 설명
 - 보통 운영체제는 사용자 스레드와 커널 스레드를 모두 제공
 - 커널은 커널 스레드만 인식
 - 스레드 구현 모델
 - * 다대일 모델
 - * 일대일 모델
 - * 다대다 모델

Slide 9

다중 스레드 모델 (계속)

Slide 10

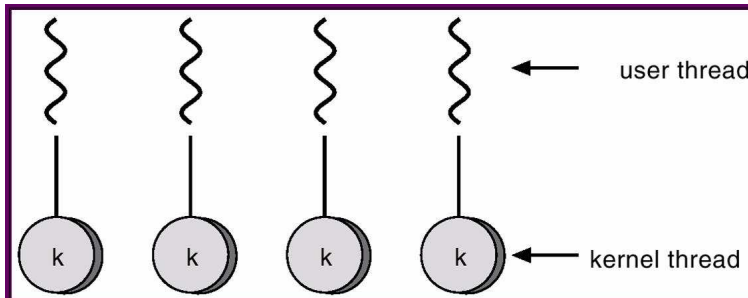
- 다대일 모델
 - 여러 사용자 스레드를 하나의 커널 스레드로 매핑
 - 장점: 스레드 처리가 빠르다
 - 단점
 - * 하나의 스레드가 block 되면 다른 스레드도 block 됨
 - * 여러개의 사용자 스레드가 다중 프로세서에서 동시에 실행될 수 없음



다중 스레드 모델 (계속)

Slide 11

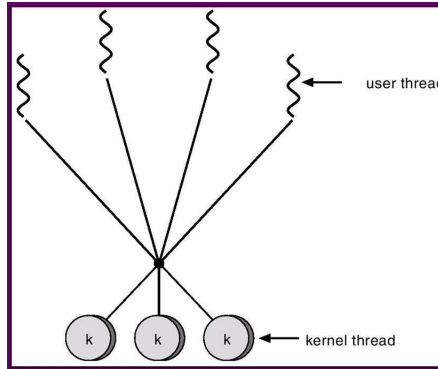
- 일대일 모델
 - 하나의 사용자 스레드가 하나의 커널 스레드로 매핑
 - 장점
 - * 하나의 스레드가 block 되어도 다른 스레드는 실행 가능
 - * 여러 사용자 스레드가 다중 프로세서에서 동시에 수행 가능
 - 단점: 생성 관리에 비용이 비쌈, 보통 스레드 수를 제한



다중 스레드 모델 (계속)

- 다대다 모델
 - 여러 사용자의 스레드를 동일 혹은 그 이하의 수의 커널 스레드로 매핑
 - 커널 스레드는 응용 프로그램 혹은 기계특성에 따라 제한 될 수 있음
 - 장점: 비용이 일대일모델보다 저렴하며 다대일 모델과 달리 다중 프로세서에서 수행 가능

Slide 12



스레드 이슈

- fork 와 exec 시스템 호출
 - 스레드를 사용할 경우 기존 fork 와 exec 를 변경해야 함
 - Unix는 두 개의 fork 시스템 호출을 제공
 1. 하나의 스레드가 fork를 호출하면 프로세스 내 모든 스레드가 복사
 2. 하나의 스레드가 fork를 호출하면 그 스레드만 복사
 - 하나의 스레드가 exec를 호출하면 프로세스 전체가 새 프로그램으로 교체
 - 어떤 fork 를 사용하는 가는 응용프로그램에 따름
 - * fork 후 바로 exec 호출 → 1번 fork 비효율, 2번 fork 사용
 - * fork 후 바로 exec 호출하지 않으면 → 1번 fork 사용해야만 됨

Slide 13

스레드 이슈 (계속)

Slide 14

- 스레드 취소 (cancellation)
 - 실행중인 스레드를 강제 종료하는 것을 의미
 - 취소할 스레드를 target thread 라 함
 - 취소하는 경우 예)
 - * 여러개의 스레드로 DB 검색중 하나의 스레드가 찾은 경우
 - * 웹페이지를 로딩중 stop 버튼이 눌러진 경우
 - 네트워크에서 데이터를 가져오는 스레드를 정지해야 함
 - 두가지 방법
 - * 비동기식 취소
 - 바로 종료 시킴
 - 자원 회수에 문제가 생길 수 있음
 - * 지연된 취소
 - 스레드 스스로 종료되어야 하는지를 주기적으로 점검하여 종료함

스레드 이슈 (계속)

Slide 15

- 신호 처리 (signal handling)
 - 유닉스에서 신호는 프로세스에게 특정 사건의 발생을 알리는 메커니즘
 - 신호는 동기식 혹은 비동기식으로 수신됨
 - * 동기식으로 수신되는 신호
 - 동기식 신호
 - 발생: 프로세스 수행에 의하여 (0으로 나누기 불법적 메모리 접근 등)
 - * 비동기식으로 수신되는 신호
 - 비동기식 신호
 - 발생: 외부에 의하여 (Ctrl C 입력등) 발생된 경우
 - 신호처리 과정
 - [단계 1] 사건의 발생으로 신호가 생성
 - [단계 2] 생성된 신호가 해당 프로세스로 전달
 - [단계 3] 프로세스는 신호를 처리

스레드 이슈 (계속)

Slide 16

- 신호 처리기 (signal handler)
 - * 기본 신호 처리기
 - 모든 신호에 대하여 기본적으로 처리하는 처리기 (커널이 수행)
 - * 사용자 정의 처리기
 - 사용자가 처리를 정의하면 기본 처리기 대신에 이것이 사용됨
- 스레드에서 신호처리 방법
 - [방법 1] 해당 스레드에게만 신호 전달 (동기식 신호처리에 적합)
 - [방법 2] 프로세스의 모든 스레드에게 신호 전달 (비동기식 신호처리에 적합)
 - [방법 3] 프로세스의 일부 스레드에게 신호 전달
 - [방법 4] 신호처리를 전담하는 스레드를 두어 처리 (Solaris 2 지원)
- 신호 처리
 - * 스레드 별로 처리할 신호와 무시할 신호 지정 가능

스레드 이슈 (계속)

Slide 17

- 스레드 풀 (thread pool)
 - 설명
 - * 프로세스 별 스레드의 수를 적절히 관리하기 위한 방법
 - * 프로세스 시작시 적정수의 스레드를 생성하여 풀에 대기
 - * 스레드 필요시 대기 풀에서 받아서 사용 후 반납
 - * 대기 풀의 모든 스레드 사용중 스레드 요청하면 사용가능할 때까지 대기해야함
 - 장점
 - * 실행 중 보다 빠른 처리 가능 (프로세스 시작시는 보다 느리지만...)
 - * 스레드 수를 제한하는 효과가 있음 (스레드 과다 생성으로 인한 문제가 발생하지 않음)
- 스레드 전용 데이터 (thread-specific data)
 - 스레드는 같은 프로세스내의 다른 스레드와 데이터를 공유
 - 스레드만 사용하는 데이터를 스레드 전용 데이터라 함
 - 대부분의 스레드 라이브러리에서 지원

스레드 실제 예

Slide 18

- Pthread
 - POSIX (IEEE 1003.1c) 표준으로서 thread 생성과 관리를 위한 API를 정의
 - API 만 정의하고 구현은 운영체제별로 다름
 - 유닉스 (Solaris 2) 에서는 Pthread 를 지원하나 윈도우는 지원하지 않음
 - Pthread 를 이용한 프로그램 예) p140 그림 5.5
- 솔라리스 2 (Solaris 2) 스레드
 - 사용자 스레드와 커널 스레드사이에 LWP 라고 하는 중간 스레드를 사용
 - 다대다 모델 사용
- 윈도우 2000 스레드
 - 일대일 모델사용
 - fiber 라이브러리를 통해 다대다 모델 기능도 제공

스레드 실제 예 (계속)

Slide 19

- 리눅스 스레드
 - Linux version 2.2 에서 스레드 도입
 - clone 시스템 호출 사용
 - * fork 와 유사하나
 - * 호출하는 프로세스를 복사하는 대신에
 - * 호출하는 프로세스와 주소공간을 공유 (마치 스레드 처럼)
- 자바 스레드
 - JVM 상에서 수행됨으로 사용자 혹은 커널 스레드로 구분이 어려움
 - 자바스레드는 JVM에 의해 커널 스레드로 매핑됨
 - 윈도우용 JVM 은 일대일 모델사용 솔라리스용 JVM 은 다대다 모델 사용