

3주차 3차시 스레드(Thread)

【학습목표】

1. 스레드에 대한 개념 설명할 수 있다.
2. 스레드의 특징에 대해 설명할 수 있다.

학습내용1 : Thread의 개요

1. Thread란?

프로세스 내에서 프로그램을 실행하는 작업 단위이다.

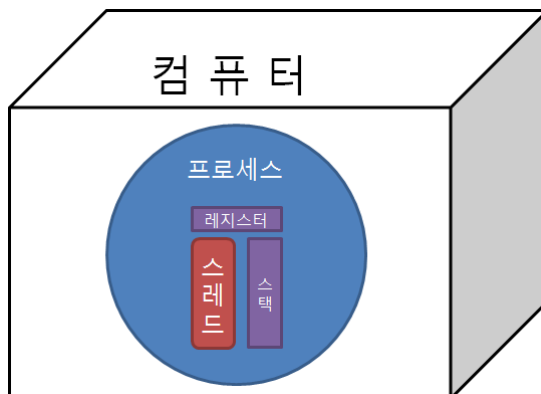
프로세스 내부에서 프로세스의 일부 기능을 갖고 있기 때문에 경량프로세스(LWP, Light Weight Process)라고도 한다.

프로세스에는 하나 혹은 여러 개의 Thread를 포함한다.

작업을 수행할 때는 Thread가 반드시 필요하다.

하나의 Thread 는 독립된 작업수행과 작업제어를 하기 위한 자기만의 스택과 레지스터를 갖는다.

2. Thread의 형상화



학습내용2 : Thread의 종류

1. 프로세스 사용방법에 따른 분류

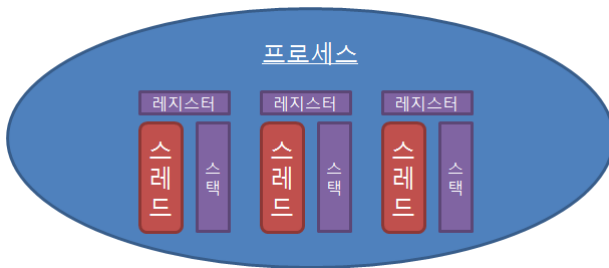
단일 Thread

- 하나의 프로세스에 한 개의 Thread 존재 경우



다중 Thread

- 하나의 프로세스에 여러 개의 Thread 존재 경우



2. 운영체제의 종류에 따른 분류

(1) 사용자 수준의 Thread

사용자가 만든 라이브러리를 사용하여 Thread 운용한다.

사용자 수준 Thread 여러 개가 커널 Thread 하나로 매핑 된다.(다대일 스레드 매핑)

속도는 빠르지만 구현이 어렵다.

문맥 교환이 적다.

독자적 알고리즘이 필요하다.

대형시스템에 적당하다.

CPU사용을 해제 못하면 시스템이 중단된다.

(2) 커널 수준의 Thread

운영체제의 커널에 의해서 Thread를 운용한다.

사용자 수준 스레드 한 개가 커널 Thread 하나로 매핑 된다.(일대일 스레드 매핑)

구현이 쉽지만 속도가 느리다.

윈도우NT/XP/2000, 리눅스, 솔라리스9 이상버전, OS/2, Mach. 등등

문맥 교환이 많다.

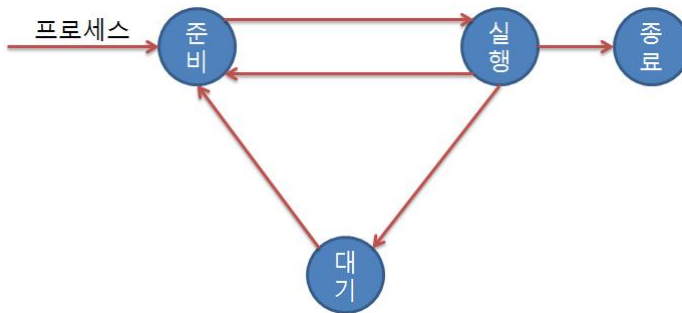
독자적 알고리즘이 필요없다.
대형시스템에 적당하지 않다.
CPU사용을 해제 못하면 운영체제가 지원한다.

(3) 혼합형 Thread

사용자 수준의 스레드와 커널 수준의 스레드를 혼합한 형태로 운용된다.
다대다 스레드 매핑
솔라리스9 미만 버전

학습내용3 : Thread의 상태

- * 하나의 CPU가 하나의 스레드를 실행한다.
- * 여러 개의 CPU가 여러 개의 스레드를 각각 동시에 처리하면 병렬처리이다.



1. 준비

스레드가 프로세스에 의해서 실행될 수 있는 상태

2. 실행

스레드가 프로세스에 의해서 실행 중인 상태 활성화 된 상태

3. 대기

Thread가 입출력 작업이 완료될 때까지 대기 상태
Thread 가 정보를 스택에 저장
같은 프로세스내의 다른 Thread 가 실행될 수 있다.

4. 종료

Thread 가 작업을 완전히 종료한 상태
레지스터와 스택을 비운다.

학습내용4 : Thread의 장점

1. Thread의 장점

- 1) 단일 프로세스를 여러 개의 스레드로 생성하여 병행성을 증진시킬 수 있다.
- 2) 하드웨어와 운영체제의 성능과 응용프로그램의 처리율을 향상시킬 수 있다.
- 3) 실행환경을 공유시켜서 기억장소의 낭비가 줄어든다.
- 4) 프로세스간의 통신 속도가 향상된다.
- 5) 공통적으로 접근 가능한 기억장치를 통해 효율적으로 통신한다.
- 6) 동일한 프로세스 환경에서 각각의 독립적인 다중 수행이 가능하다.
- 7) 스레드 기반 시스템에서 스레드는 독립적인 스케줄링의 최소단위로써 프로세스의 역할을 담당한다.

【학습정리】

1. 스레드는 프로세스 내에서 프로그램을 실행하는 작업단위이다.
2. 스레드의 종류(프로세스 사용방법)
 - 단일 스레드
 - 다중 스레드
3. 스레드의 구분(운영체제의 종류)
 - 사용자 수준의 스레드
 - 커널 수준의 스레드
 - 혼합형 스레드