컴퓨터 네트워크

**스레드 조사**

**.**

|  |  |
| --- | --- |
| **학과** | **컴퓨터 공학과** |
| **학번** | **2015040044 , 2015040044** |
| **이름** | **전승원 , 신영호** |
| **제출일** | **2019. 04. 02** |

**프로세스와 스레드의 정의**

**프로세스란** 사전적으로 “컴퓨터에서 연속적으로 실행되고 있는 컴퓨터 프로그램”을 의미합니다. 프로세스는 메모리에 올라와 실행되고 있는 프로그램의 독립적인 개체이자 운영체제로부터 시스템 자원을 할당 받는 작업의 단위 즉, 동적인 개념으로는 실행된 프로그램을 의미합니다.

**스레드란** 사전적으로 “프로세스 내에서 실행되는 여러 흐름의 단위”를 의미합니다. 프로세스의 특정한 수행 경로이며 프로세스가 할당 받은 자원을 이용하는 실행의 단위입니다.

**프로세스와 스레드의 차이점**

**프로세스는** 각각 독립된 메모리 영역(Code, Data, Stack, Heap의 구조)을 할당 받으며, 기본적으로 프로세스당 최소 1개의 스레드를 가지고 있습니다. 또한 각 프로세스는 별도의 주소 공간에서 실행되며, 한 프로세스는 다른 프로세스의 변수나 자료구조에 접근할 수 없습니다. 한 프로세스가 다른 프로세스의 자원을 접근하려면 프로세스 간의 통신을 사용해야 합니다.

**스레드는** 프로세스 내에서 각각 Stack만 따로 할당 받고 Code, Data, Heap 영역은 공유합니다. 프로세스는 다른 프로세스의 메모리에 직접 접근할 수 없는 반면에 스레드는 한 프로세스 내에서 동작되는 여러 실행의 흐름으로, 프로세스 내의 주소 공간이나 자원들(Heap 공간 등)을 같은 프로세스 내에 스레드끼리 공유하면서 실행됩니다. 각각의 스레드는 별도의 레지스터와 스택을 갖고 있지만, 힙 메모리는 서로 읽고 쓸 수 있습니다.

**스레드의 사용 이유**

한 프로세스에 포함된 스레드들은 공통의 목적 달성을 위해 병렬로 수행을 하게 됩니다. 따라서, 자원을 공유하여 한 프로세스에서 동시 작업이 가능하게 됩니다. 또한 스레드를 사용하면 하나의 프로세스가 서로 다른 프로세서에서 프로그램의 다른 부분을 동시에 실행이 가능하게 됩니다. 즉, 응용 프로그램 하나가 비슷한 작업들을 여러 개 수행이 가능합니다. 이것으로 시스템 성능과 효율을 향상을 시킬 수 있습니다.

스레드를 이용하여 응용 프로그램의 일부분이 봉쇄 또는 긴 작업 수행시에도 프로그램 실행을 계속 허용하여 사용자에 대한 응답성을 증가시킬 수 있습니다. 스레드는 한 프로세서에서 자원을 공유하므로 프로세스를 생성하는 것보다 오버헤드를 줄여 좋은 경제성을 얻을 수 있습니다. 그리고 다중 프로세서 구조에서도 활용이 가능합니다.

**스레드 모델**

**스레드를 위한 지원은 사용자 스레드와 커널 스레드가 있습니다.**

사용자 스레드는 커널 위에서 지원되며 커널의 지원 없이 관리가 됩니다. 커널 스레드는 운영체제에 의해 직접 지원되고 관리되는 스레드입니다.

**스레드의 모델에는 3가지가 있습니다.** 다대일 모델, 일대일 모델, 다대다 모델이 있습니다

**다대일 모델은** 많은 사용자 수준 스레드를 하나의 커널 스레드로 매핑 하는 것입니다. 스레드 관리는 사용자 공간의 스레드 라이브러리에 의해 행해집니다. 스레드 관리는 효율적이지만 한 스레드가 긴 작업을 실행할 경우 나머지 작업들은 대기하게 됩니다. 즉, 다중 CPU환경에서도 병렬로 작업을 처리할 수 없게 됩니다.

**일대일 모델은** 각 사용자 스레드를 각각 하나의 커널 스레드로 매핑 하는 것입니다. 하나의 스레드가 긴 작업을 처리하는 동안에도 다른 스레드가 실행될 수 있기 때문에 다대일 모델보다 더 많은 병렬성을 제공합니다. 이 모델의 단점은 사용자 수준 스레드를 생성할 때 그에 따른 커널 스레드를 생성해야 한다는 것입니다. 커널 스레드를 생성하는 오버헤드로 응용프로그램의 성능저하를 유발할 수도 있습니다.

**다대다 모델은** 여러 개의 사용자 스레드를 그보다 작거나 같은 수의 커널 스레드로 다중화 하는 것입니다. 커널 스레드의 수는 응용 프로그램이나 특정 기계에 따라 결정됩니다. 응용 프로그램은 단일 처리기보다 다중 처리기에 더 많은 커널 스레드를 할당 받을 수 있습니다. 개발자는 필요한 만큼 많은 사용자 수준 스레드를 생성할 수 있으며 그에 상응하는 커널 스레드가 다중 처리기에서 병렬로 수행됩니다. 스레드가 긴 작업등의 호출을 발생시켰을 때 커널이 다른 스레드의 수행을 스케줄링 합니다.

**Python에서 프로세스 및 스레드를 사용하는 방법**

**프로세스의 사용은** multiprocessing 모듈에서 Process클래스를 사용하여 할 수 있습니다. (from multiprocessing import Process)

Process(target=, args=, kwargs=, daemon=None)으로 이루어져 있으며 target에는 프로세스에 디스패치할 함수가 들어가고 args에는 target함수에 튜플 형태로 넘겨질 인자 값이, kwargs에는 사전의 형태로 인자 값이 들어갑니다. deamon에는 해당하는 프로세스가 데몬으로 실행될지의 여부가 들어갑니다.

프로세스를 시작하려면 start() 메소드를 이용하여 실행하고 이 메소드가 호출되면 자신의 run() 메소드를 호출합니다. Join() 메소드를 이용하여 프로세스가 끝날 때까지 기다릴 수 있습니다. 그외에 프로세스의 이름 속성 name과 데몬 속성 daemon이 있습니다.

**스레드의 사용은** threading 모듈에서 Thread클래스를 사용하여 할 수 있습니다.

(from threading import Thread)

Thread(target=, args=, kwargs=, daemon=None)으로 이루어져 있으며 속성들과 메소드들은 프로세스의 사용 방법과 동일합니다.