

● Multi thread에 대해 설명해라.

▶ thread는 한 process 내에서 실행되는 동작, 즉 function의 단위입니다. Multi thread는 한 프로세스 내에 여러 개의 thread가 있는 것으로, 각 thread가 프로세스 내에서 독립적인 기능을 수행함으로써 하나의 프로세스가 동시에 여러 작업을 병렬로 처리할 수 있습니다. 각 thread들은 속해 있는 프로세스의 Stack 메모리를 제외 한 나머지 메모리 영역 — Code, Data, Heap 영역을 공유합니다.

● thread는 왜 독립적인 Stack memory 영역이 필요한가?

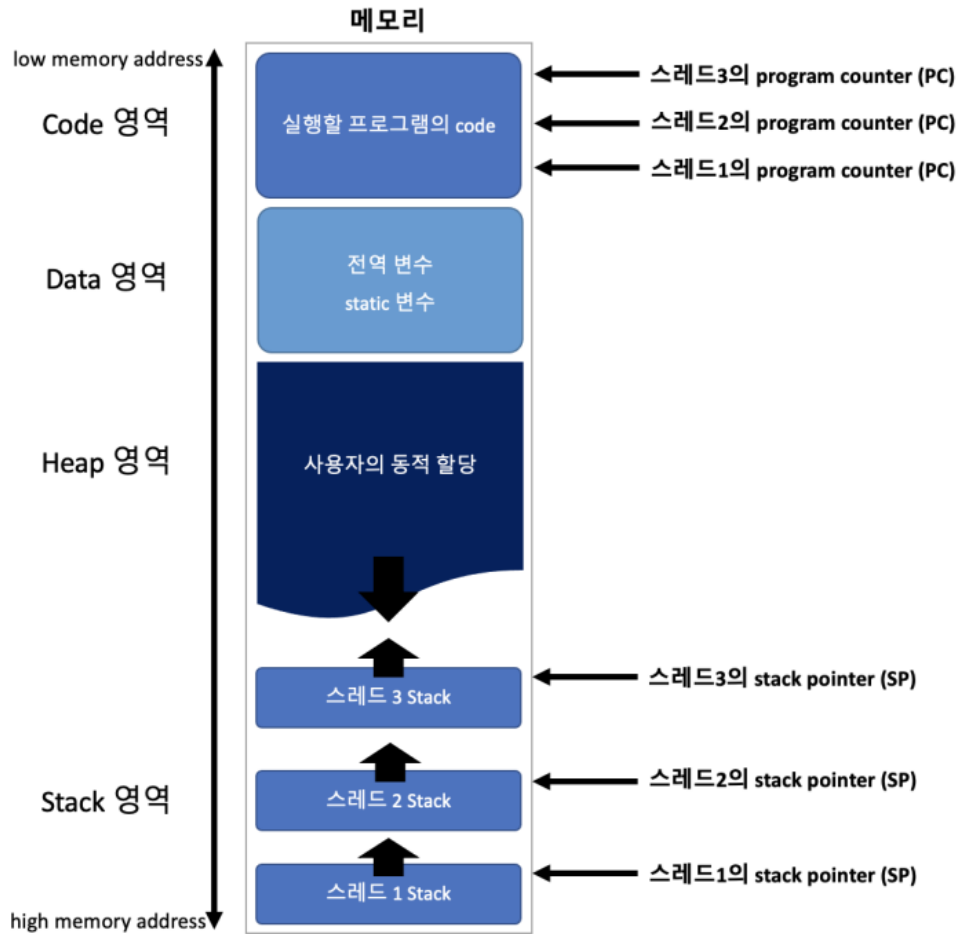
▶ Stack 영역은 함수 호출 시 전달되는 인자, 함수의 Return address, 함수 내 지역 변수 등을 저장하기 위한 메모리 영역입니다. thread가 프로세스 내에서 독립적인 기능을 수행한다는 것은 독립적으로 함수를 호출함을 의미합니다. 따라서 각 thread가 독립적인 동작을 실행하기 위해서는 각 thread의 stack memory 영역이 독립적이어야 합니다. 결과적으로 thread는 process로부터 Stack memory 영역은 독립적으로 할당받고, 나머지 영역 — Code, Data, Heap 영역은 공유하는 형태를 갖게 됩니다.

● multi thread and Program Counter (PC)

※ multi thread, multi process 환경에서는 PC 값을 PCB(process control block)가 아닌 TCB(thread control block)에 저장합니다.

▶ multi thread에서는 각각의 thread가 자신의 PC값을 TCB(Thread Control Block)에 저장을 합니다(multi thread에서는 각각의 thread가 독립적인 PC값을 가지고 있어야 합니다). 그 이유는 한 process 내에서도 thread 사이에 context switching이 일어나는데, TCB에 code address(PC)가 저장되어 있어야 해당 스레드가 어느 코드 영역까지 진행했는지를 기억하여 그 다음 코드부터 실행을 할 수 있기 때문입니다. 예를 들면 thread a 에서 thread b로 context switch가 일어나면, 현재의 PC register에 저장되어 있는 값을 thread a의 TCB에 저장을 하고, thread b의 TCB의 PC값을 불러와 PC register에 저장을 합니다. CPU는 PC register가 새롭게 가리키는 thread b의 code영역, 즉 thread b 내에서 다음에 실행해야 할 명령의 주소값(PC)을 참조하여 실행을 하게 되는 것입니다.

▶ multi thread concurrency: thread 사이에 context switching을 하다가, 또한 process 사이에서도 context switching을 하는 것입니다. 예를 들면 process 1과 process 2에 각각 thread a, thread b가 있다고 하면, process 1의 thread a에서 process 2의 thread b로 context switching이 일어나면, PC register의 값을 process 1 thread a의 TCB에 저장을 하고, process 2 thread b의 TCB의 PC값을 불러와 PC register 값에 저장을 하면, CPU는 PC register가 가리키는 process 2 thread b의 code 영역을 참조하여 실행을 하게 되는 것입니다.



● process와 thread를 비교해서 설명해라

▶ process는 운영체제로부터 자원을 할당받는 작업의 단위이고, thread는 process가 할당받은 자원을 이용하여 실행하는, 실행의 단위입니다. 즉, process는 실행파일 program이 memory에 적재되어 CPU를 할당받아 실행되는 것이고, thread는 한 process 내에서 실행되는 동작의 단위인 것입니다. 각 process에 할당되는 memory 공간은 code, data, heap, stack 네 가지의 영역이 있는데, thread는 process내에서 stack영역을 제외한 code, data, heap영역을 공유합니다.

Resources: inflearn 개발남 노씨