

프로세스와 스레드

프로세스

- 실행중인 프로그램
- 가상주소 공간을 가지고 있고 운영체제가 실행하고 스케줄링한다. 이 특징으로 인해 운영체제가 각각의 프로세스를 독립적으로 다룰 수 있다.
- 즉 운영체제로부터 자원을 할당받은 작업단위

쓰레드

- 프로세스에서 실행되는 명령어들의 유닛
- 프로세스가 할당받은 자원을 이용하는 실행 흐름의 단위

프로세스의 구성

PCB

프로세스의 정보는 PCB(Process Control Block)에 저장됩니다.

- PCB는 프로세스의 여러가지 정보를 가지고 있는데, 프로세스 ID, 현재 상태, 큐에서의 우선순위, 프로세스가 메모리 어디에 저장되어 있는지, context 데이터들의 위치 등등을 포함하고 있습니다.
- OS에 의해 생성되고 관리된다. Primary Process Table에 PCB의 리스트들이 존재한다.
- 프로세스당 한개씩 가지고 있다.

Identifier
State
Priority
Program counter
Memory pointers
Context data
I/O status information
Accounting information
⋮

프로세스의 메모리 공간

PCB에서 프로세스가 메모리 공간 중에 어디에 위치하는 정보가 담겨있다고 했는데, 이 메모리는 어떻게 구성되어 있는지 살펴봅시다.

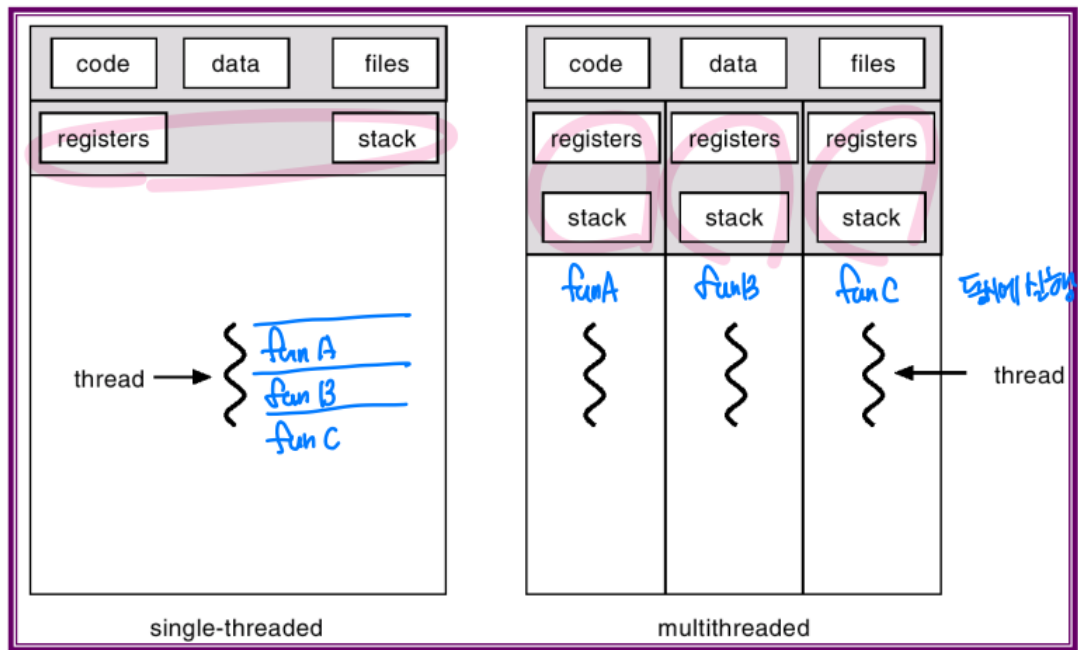
이 공간은

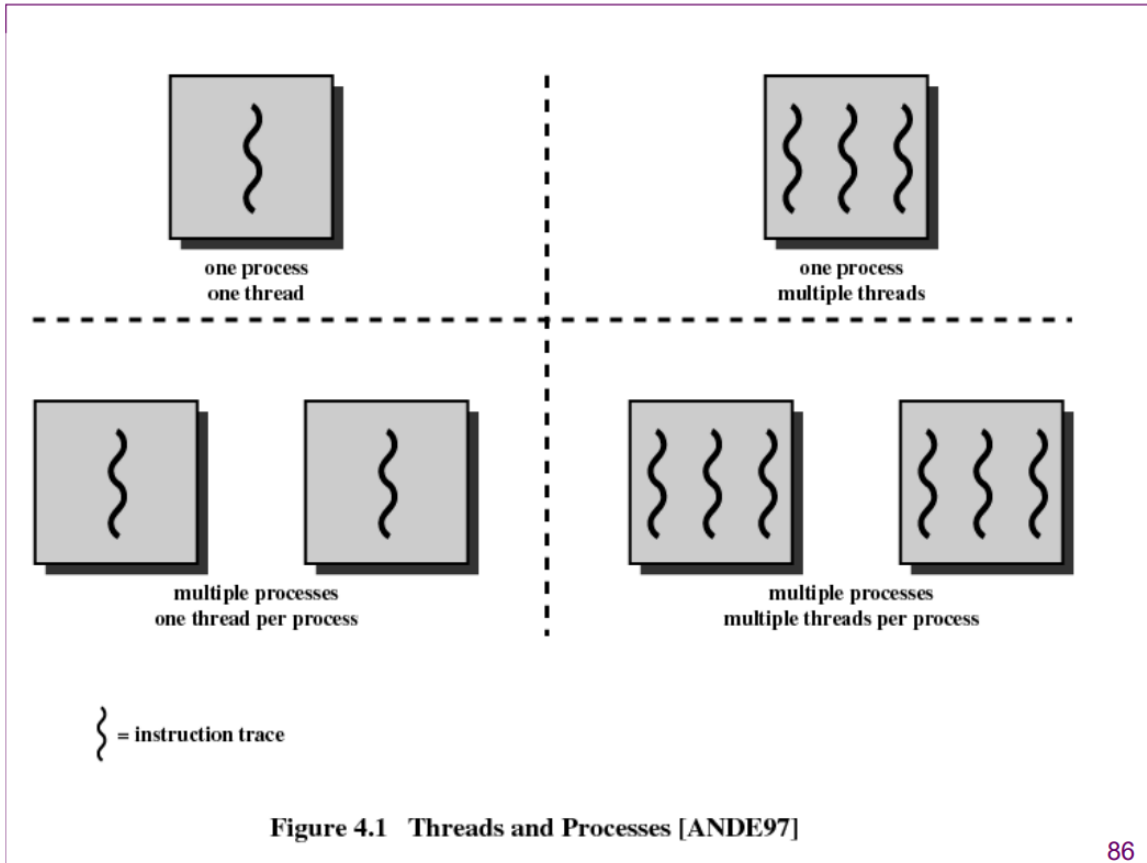
- Code
- Data
- Stack
- Memory for registers context

로 이루어져 있고 이중 Stack와 registers 한쌍이 한개의 스레드가 가지게 된다.

멀티스레드

- 동시에 동작하고 있는 스레드로 구성되어진 프로세스를 의미
- 스레드는 속한 프로세스의 메모리와 자원들에 접근할 수 있다. 즉 같은 프로세스에 속한 스레드는 자원들을 공유하고 있다.
- 스레드는 지역변수를 저장하기 위한 정적 저장소(stack)를 스레드별로 가지고 있기도 한다.





86

문맥전환

Context : 프로그램의 실행되던 환경

Context Swith : 프로세스에서 다른 프로세스로 CPU의 제어권을 넘기는 과정

1. 프로그램 카운터와 다른 레지스터들을 포함한 프로세서의 context를 저장한다.
2. 현재 실행중인 프로세스의 pcb를 업데이트한다.
3. 적절한 큐로 pcb를 옮긴다.
4. 실행한 프로세스를 고른다
5. 고른 프로세스의 pcb를 업데이트한다
6. 고른 프로세스의 context를 저장한다.

쓰레드의 장점

- 프로세스보다 새로운 쓰레드를 만드는데 더 적은 시간이 걸린다.
- 프로세스보다 쓰레드를 terminate하는데 더 적은 시간이 걸린다.
- 같은 프로세스 내의 두개의 쓰레드 사이에서 전환속도가 빠르다
- 프로세스끼리 통신하는 것보다 쓰레드 사이에서 통신하는 것이 더 빠르다
- 쓰레드는 메모리와 파일들을 공유할 수 있기때문에 그들끼리 통신할때 커널을 통하지 않아도 된다.

퀴즈

1. 프로세스는 실행중인 프로그램이라고 하였는데, 프로세스와 프로그램은 어떤 차이가 있는지?
2. 프로세서는 무엇이며, CPU는 무엇일까?
3. 운영체제가 프로세스를 만든다고 하였는데, 여기서 운영체제는 무엇을 의미하는가?