

네트워크 프로그래밍

12. 다중 접속 서버 구현

다중 접속 서버 구현 방법

- **멀티프로세스(Multiprocess) 기반 서버**
 - ▣ 다수의 프로세스를 생성하는 방식
- **멀티플렉싱(Multiplexing) 기반 서버**
 - ▣ 입출력 대상을 묶어서 관리하는 방식
- **멀티쓰레드(Multithread) 기반 서버**
 - ▣ 클라이언트의 수만큼 쓰레드를 생성하는 방식

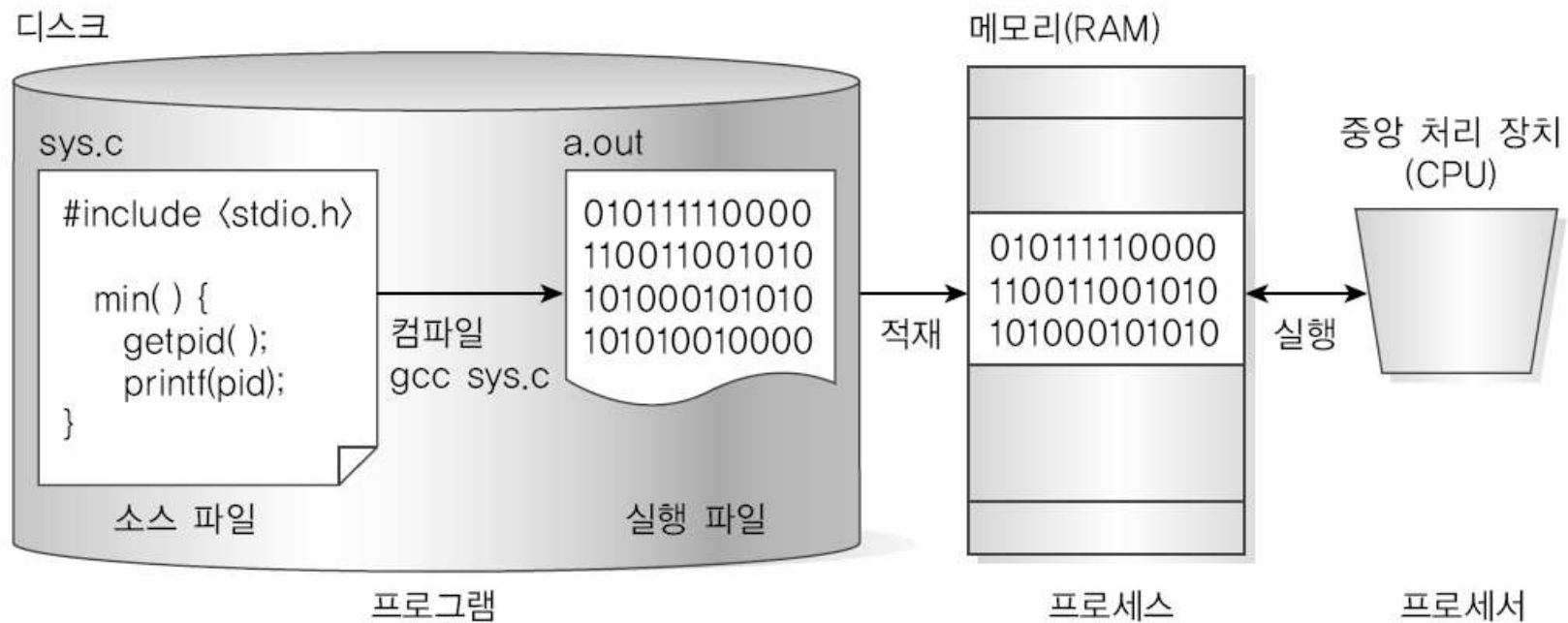
프로세스

□ 프로세스란?

- ▣ 실행중인 프로그램
- ▣ 프로그램은 디스크에 저장된 파일과 같은 수동적인 개체
- ▣ 프로세스는 프로그램 카운터(PC)를 가진 능동적인 개체
- ▣ 프로그램이 주기억장치에 적재되어 처리를 시작할 때 비로소 프로세스라는 활동적인 개체가 됨

프로그램과 프로세스의 관계

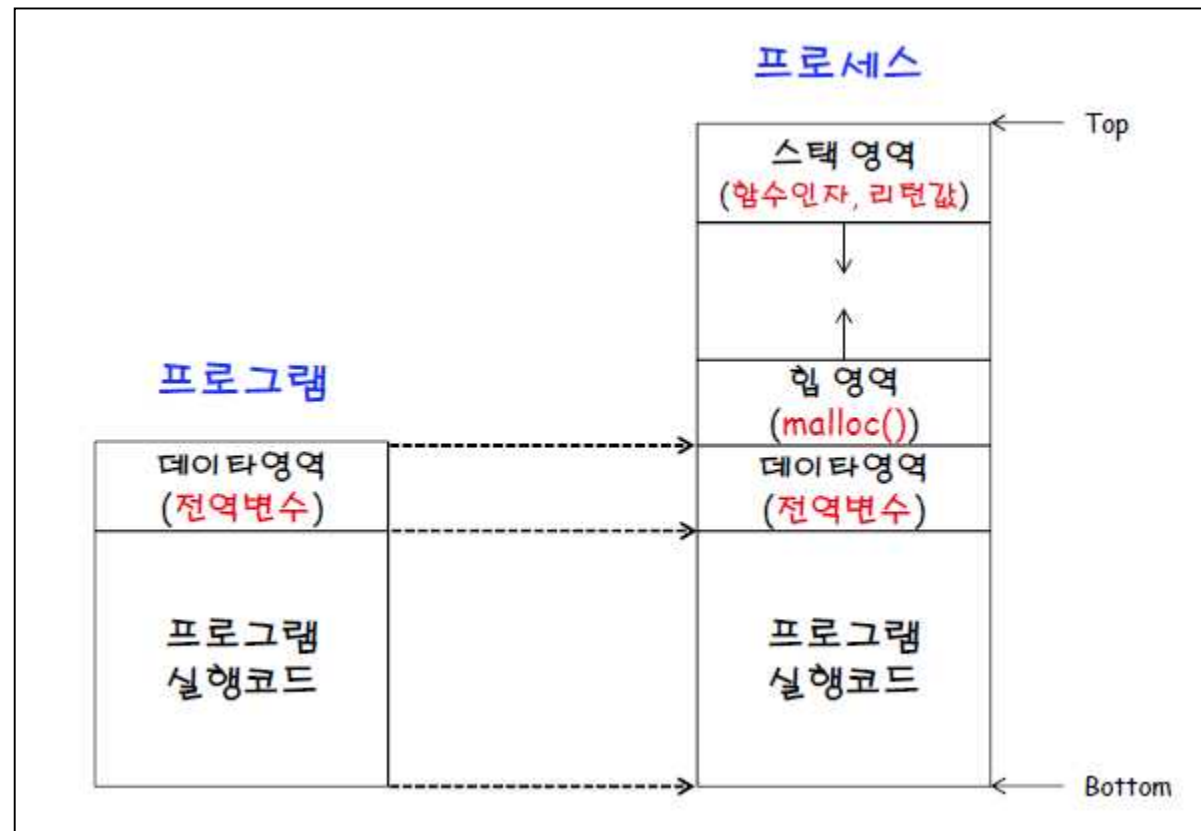
- 프로그램(program): 사용자가 컴퓨터에 작업을 시키기 위한 명령어의 집합
- 프로세스(process): 실행중인 프로그램
- 프로세서(processor): 중앙처리장치



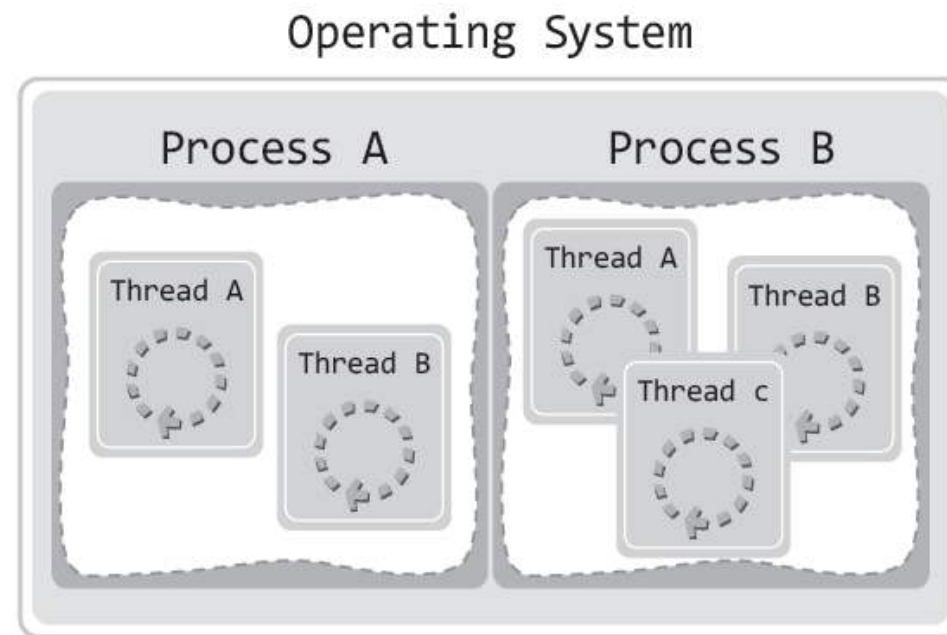
실행 파일 구조 & 프로세스의 구조

메모리에 로드된 프로그램을 프로세스라고 한다.
메모리에 로드되어야 스택, 힙영역이 생긴다

질문: 스택과 힙은 비슷한 공간을 사용하는 것이 아닌가. 그래서 오버플로우가 발생하는거 아닌가요?



프로세스와 스레드의 관계

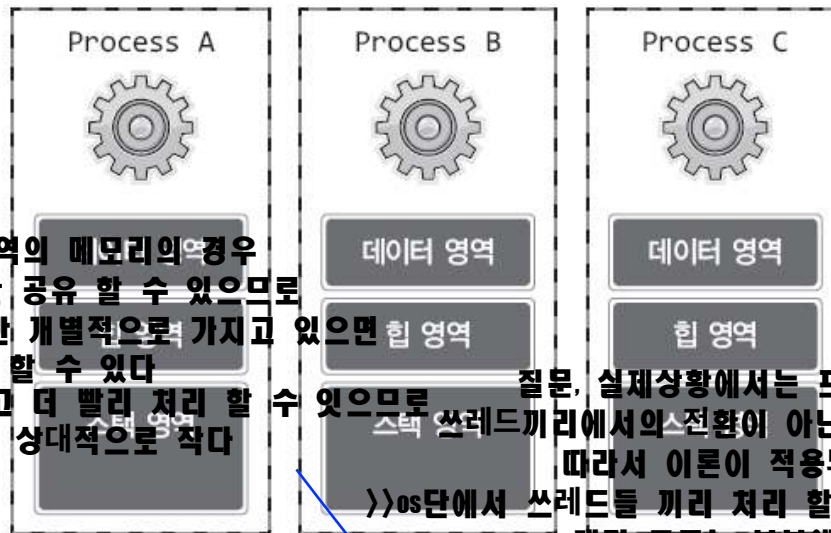


PCB란 os가 프로세스를 표현한것,
즉, 프로세스에 대한 메타데이터이다

멀티프로세스 vs 멀티쓰레드

쓰레드가 실행하다가 cpu에서 쫓겨나면
> 다시 cpu로 들어오게 될 경우
어디까지 했는지 저장하는 작업이 필요
>> PCB(process control block)가 맡음
또한 이제 다른 쓰레드로 들어 갈 경우 그 쓰레드의
문맥을 찾아서 거기서 부터 실행 해야 함!

멀티 프로세스(Multiprocesses)



- 운영체제 관점에서의 실행흐름을 제어한다.
- 컨텍스트 스위칭에 대한 부담이 크다
- 프로세스간 데이터 교환이 불가능하다. (IPC 필요)

* 컨텍스트 스위칭(context switching): 현재 실행중인
프로세스의 실행 정보를 PCB에 저장하고 다음으로
실행할 프로세스의 실행 정보를 가져오는 과정

멀티 쓰레드 (Multithreads)



- 프로세스 내에서의 실행흐름을 제어한다.
- 컨텍스트 스위칭에 대한 부담이 덜하다.
- 쓰레드간 데이터 교환이 매우 쉽다.

멀티프로세스 (Multiprocess)

□ 구조

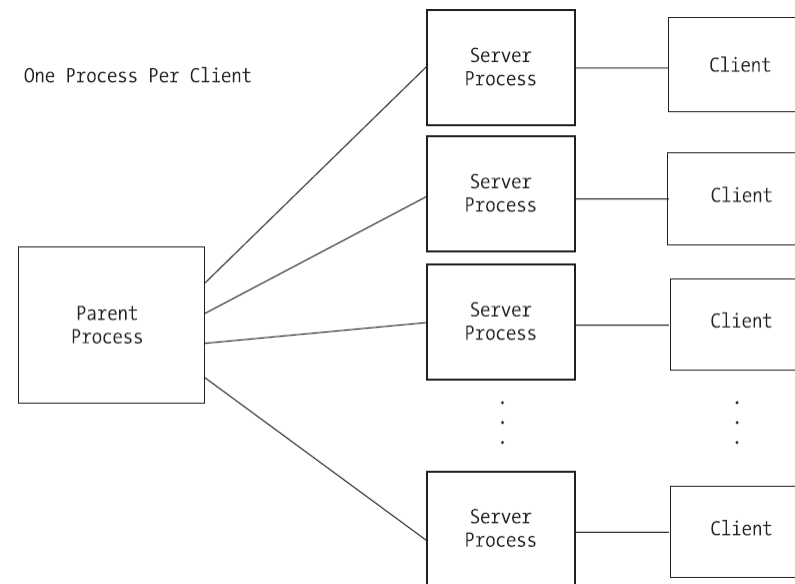
- 클라이언트 마다 서버 프로세스를 생성하는 구조

□ 장점

- 구현이 단순하다.
- 안정성이 높다.
- 다수의 요청을 동시에 처리 가능

□ 단점

- 프로세스 생성 오버헤드가 발생
- 많은 자원을 소요한다.



멀티플렉싱 (Multiplexing)

□ 구조

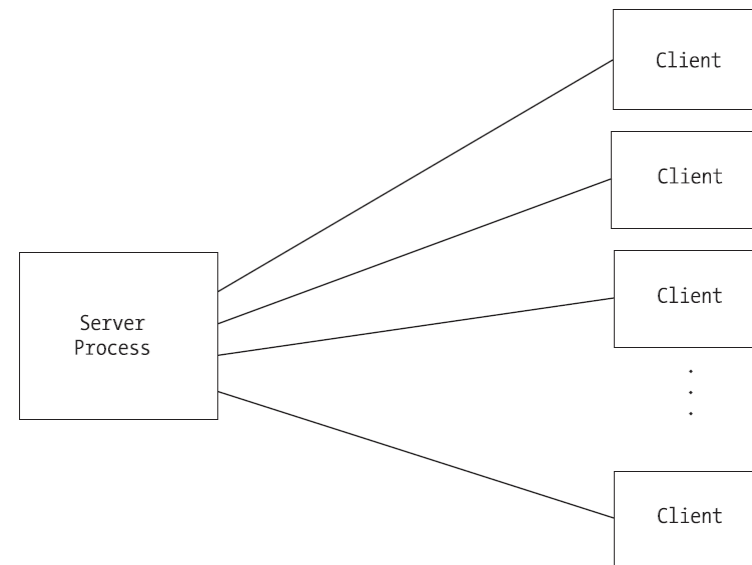
- 한 프로세스가 다수의 클라이언트를 처리할 수 있는 구조

□ 장점

- 구현이 단순하며 짧은 연결에 유용하다.
- 프로세스 생성 오버헤드가 없다.

□ 단점

- 프로세스가 다운될 경우 안정성이 낮다.
- 한번에 여러 요청을 처리하지 못한다.
- 접속허용량이 낮다.



멀티쓰레드 (Multithreads)

□ 구조

- 쓰레드 당 하나의 클라이언트를 처리하는 구조

□ 장점

- 멀티프로세스에 비해 비교적 자원 소요량이 적다.
- 다수의 요청을 동시에 처리 가능
- 구현이 단순하다.

□ 단점

- 프로세스 다운시 안정성이 낮다.

