# Chapter 12. POSIX thread

- 적은 오버헤드로 할수있게해서 concurrent한 실행
  - asynchronous한 event를 효율적으로 처리.
  - parallel performance

- 각 스레드는 하나의 실행 흐름. (stack, CPU state, registers를 각각 가짐.)

- 서로 공유하는 부분: 코드부분, 전역변수, 힙메모리 등 공유하는 메모리가 존재.
  ↓
  여전히 다중프로세스는 OS 이용해서만 통신할 수 있는 점과 다름.

- 멀티 태스킹
  Single CPU 에서도 time-division multiplexing으로 멀티태스킹 가능.
  문맥 바꿔치기 스위칭.
  multi 프로세서 CPU에서는 실제로 동시에 멀티태스킹.

| process | vs | thread |
|---|---|---|
| independent | | 서로 dependent한 부분 있음. |
| 더 많은 state info | | |
| 서로 보호된 메모리 주소 | | |
| OS가 제공하는 IPC 메커니즘으로 정보공유 | | |
| 운영체제에 따른 멀티태스킹 | | |

## pthread

- pthread_self (void) 자기 id 반환

- pthread_equal (t1, t2) 같으면 nonzero. 아니면 0.

- pthread_create (thread_id, attr, void start_routine(void), arg)
  - thread_id: 생성된 스레드 정보
  - start_routine: 새로 만들어질 스레드가 실행할 함수

- pthread_detach (thread) detach 되면, thread 종료시 바로 리소스 반납
  종료 상태도 반환 X

pthread_join (thread, value_ptr)
- value_ptr: 스레드 종료후 반환 값

- pthread_exit (value_ptr) 또는 스레드 안에서 return

- pthread_cancel (thread) 종단요청 보내기

- pthread_setcancelstate ( state_enable_or_disable, old_state)